

CubeSLAM: Monocular 3D Object Detection and SLAM without Prior Models

Shichao Yang, Sebastian Scherer

Robotics Institute, Carnegie Mellon University
`{shichaoy, basti}@andrew.cmu.edu`

Abstract. We present a method for single image 3D cuboid object detection and multi-view object SLAM without prior object model, and demonstrate that the two aspects can benefit each other. For 3D detection, we generate high quality cuboid proposals from 2D bounding boxes and vanishing points sampling. The proposals are further scored and selected to align with image edges. Experiments on SUN RGBD and KITTI shows the efficiency and accuracy over existing approaches. Then in the second part, multi-view bundle adjustment with novel measurement functions is proposed to jointly optimize camera poses, objects and points, utilizing single view detection results. Objects can provide more geometric constraints and scale consistency compared to points. On the collected and public TUM and KITTI odometry datasets, we achieve better pose estimation accuracy over the state-of-the-art monocular SLAM while also improve the 3D object detection accuracy at the same time.

1 Introduction

3D object detection and Simultaneous localization and mapping (SLAM) are two important tasks in computer vision and robotics. For example in autonomous driving, vehicles need to be detected in 3D space in order to keep safe. In augmented reality, 3D objects also need to be localized for more realistic physical interactions. Different sensors can be used for these tasks such as laser-range finders, RGB-D or stereo cameras which can directly provide depth measurement. Monocular camera also attracts many interests due to its low cost and small size. Most of the existing monocular approaches solve object detection and SLAM separately and also depend on the prior object CAD models which maybe not applicable for general environments. Therefore in this work, we focus on 3D object mapping without priors and solve 3D object detection and multi-view object SLAM jointly.

For object detection, many algorithms are able to detect different 2D objects with various size and viewpoints in large datasets using convolutional neural network (CNN) [1]. 3D object objection also attracts more attention recently especially for robotic applications [2]. For the SLAM or Structure from Motion (SfM), the classic approach is to track visual geometric features such as points

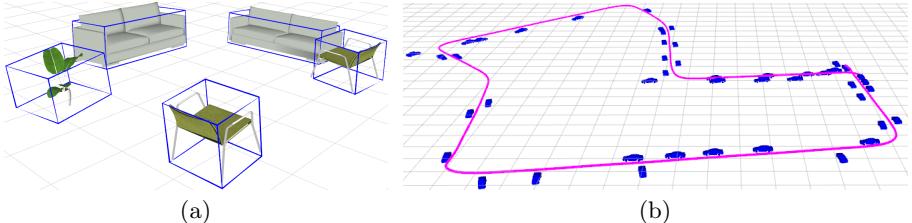


Fig. 1. Monocular 3D object detection and mapping without prior object models. Mesh model is just for visualization and not used for detection. (a) ICL NUIM data with various objects, whose position, orientation and dimension are optimized by SLAM. (b) KITTI 07. With object constraints, our monocular SLAM can build a consistent map and correct scale drift, without loop closure and constant camera height assumption.

[3], lines [4], planes across frames then minimize the reprojection or photometric error through bundle adjustment (BA). However, apart from these low-level features, objects also play important roles in the environments which have not been explored much in SLAM. Detecting and mapping 3D objects can greatly improve robot intelligence for environment understanding and human-robot interaction. In addition, objects as SLAM landmark can also provide additional semantic and geometric constraints to improve camera pose estimation.

In this work, we propose a system to solve 3D object detection and SLAM together. Given the 2D object detection, we efficiently generate many 3D cuboid proposals through vanishing points (VP) sampling, under the assumptions of fitting tightly with 2D bounding box after projections. Then the selected cuboid proposals are further optimized with points and cameras in multi-view BA. Objects are utilized in two folds: provide depth initialization for points difficult to triangulate and provide geometry constraints in BA. The estimated camera poses from SLAM can improve the single-view object detection as well. In summary, our contributions are as follows:

- Propose an efficient and accurate 3D cuboid fitting approach for single image, without prior object model or prior orientation and dimension prediction.
- Propose an object SLAM with novel measurements between points, cameras and objects, achieving better accuracy on many datasets including KITTI. Object detection and SLAM are demonstrated to benefit each other.

In the following, we first introduce the single image detection in Sec 3, then explain the object SLAM in Sec 4, followed by experiments in Sec 5.

2 Related Work

Single image 3D object detection Existing approaches can be divided into two categories, with or without prior CAD models. With prior model, the best object pose to align with RGB images can be found through hand-crafted features [5] or more recent deep networks [6] [7] [8]. Without prior models, the

early approaches focus on the geometry modelling. For example, objects can be generated by a combination of Manhattan edges or rays through VPs [9] [10]. Chen *et al.* proposed to exhaustively sample many 3D boxes on the ground then select based on various context features [11]. One similar work to us is [12] which used projective geometry to find cuboids to fit tightly with 2D bounding box. We extend it to work without prediction of object size and orientation.

Multi-view object SLAM There are many point based visual SLAM algorithms such as ORB SLAM [3], and DSO [13], which can achieve impressive results in general environments. Object augmented mappings are also approached in recent years. There are typically two categories of them, either decoupled or coupled. The decoupled approaches first build SLAM point cloud map then further detect and optimize 3D object poses based on point cloud clustering and image evidence filtering [14] [15] [16]. It shows improvement compared to 2D object detections but it doesn't change SLAM part thus may not work well if SLAM cannot build high quality map.

The coupled approach is usually called object level SLAM. Bao *et al.* proposed the fist Semantic SfM to jointly optimize camera poses, objects, points and planes [17]. [18] proposed a practical SLAM system called SLAM++ using RGB-D cameras and prior CAD models. Frost *et al.* represented object as spheres to correct the scale drift of monocular SLAM [19], similarly in [20]. Recently, [21] proposed a real time monocular object SLAM using prior database. [22] solved SfM theoretically with ellipsoid object and affine cameras. Uncertain data association of object SLAM is addressed in [23][24]. [25] proposed similar idea to combine scene understanding with SLAM but only applied to planes.

3 Single Image 3D Object Detection

3.1 3D proposal generation

A general 3D cuboid can be represented by 9 DoF parameter: 3 DoF position, 3 DoF rotation and 3 DoF dimension. The cuboid coordinate frame is built at the cuboid center, aligned with the main axes. The camera intrinsic calibration K is also known. Based on the assumptions that the cuboid's projected corners should fit tightly with 2D bounding box, there are four constraints corresponding to four sides of a rectangle which cannot fully constrain all 9 parameters. Thus other information is required for example the provided or predicted object dimensions and orientations, used in vehicle detection [12] [11]. Different from them, we utilize the VPs to change and reduce the regression parameters in order to work for general objects.

As we know, 3D cuboid has three orthogonal axes and can form three VPs after perspective projections [10] depending on object rotation R and camera calibration K . Therefore we can estimate the object's 3 DoF rotation R to compute VPs, sample one corner on the rectangle's top edge, then all the other seven corners can be computed analytically. As shown in Fig. 2, we select three most common cuboid configurations with different observable faces to demonstrate

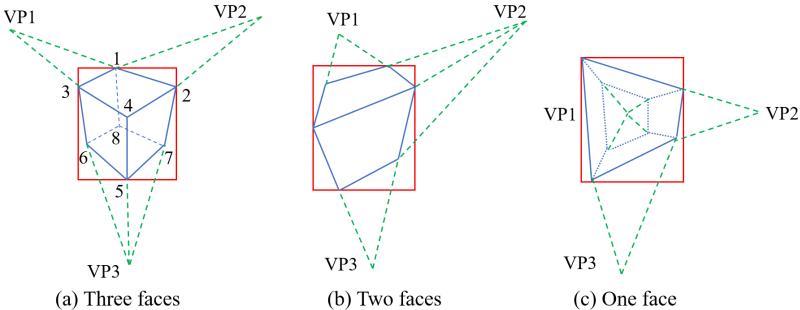


Fig. 2. Proposals generation from 2D object box. Cuboids are divided into three categories depending on the number of observable faces. If one corner is estimated, the other seven corners can also be computed from vanishing points (VPs). For example in (a), if corner 1 is sampled, then corner 2 and 3 can be determined through ray intersection of VP line and rectangles, followed by corner 4 and other bottom corners.

this process. After we get 8 cuboid corners in 2D, we can backproject to 3D space to compute 3D position and dimensions which is up to a scale factor. The scale can be reasoned from camera height to ground, prior object size and so on.

Summary We first estimate the object 3 DoF rotation and one corner on the top boundary, then compute all other 2D corners and further back-project to get 3D position and dimensions. The problem now changes to how to get object rotation and top corner. Though deep networks can be used to directly predict them with large data training, we choose to sample exhaustively then score to select the best in order to fully utilize geometry constraints from latter SLAM optimization. For objects lying on the ground, the object rotation to compute VP can also be represented by object yaw and camera roll/pitch *wrt.* ground. More importantly, camera roll/pitch can be estimated from SLAM or other sensors to greatly reduce sampling space. For example on the KITTI dataset, the camera is nearly parallel to ground with fixed roll/pitch and fixed camera height.

3.2 Proposal scoring

After sampling many cuboid proposals, we define cost functions to score them shown in Fig. 3. Different cues have been proposed before such as semantic segmentation [11], edge distance [5], HOG features [9]. We propose some fast and effective cost functions to best align the cuboid with image edges. This approach works the best for "boxy with clear edge" objects, but also works decently well for bicycles and toilet etc. shown in later experiments due to constraints from VP and 2D box. In addition, SLAM will also post-optimize to improve 3D object accuracy, which is the key motivation of this paper. We can denote the image as I and cuboid proposal as x , then the cost function is defined as:

$$E(x|I) = \phi_{dist}(x) + w_1\phi_{angle}(x) + w_2\phi_{shape}(x) \quad (1)$$

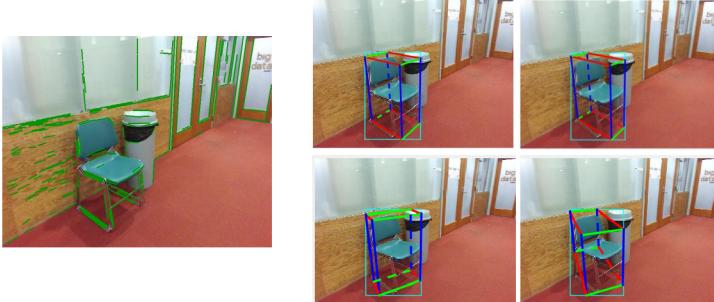


Fig. 3. Cuboid proposal scoring. (**Left**) Edges to align and score the proposals. (**Right**) Cuboid proposals generated from the same 2D cyan bounding box. The top left is the best and bottom right is the worst after scoring.

where $\phi_{dist}, \phi_{angle}, \phi_{shape}$ are three costs explained as follows. w_1 and w_2 are weight parameters and set to be $w_1 = 0.7, w_2 = 2.5$ after manual search on small sample datasets.

Distance error ϕ_{dist} The 2D cuboid edges should match with the actual image edges. We use Canny edge detection to build distance map, then sum over the cuboid edge's Chamfer distance, normalized by 2D box size.

Angle alignment error ϕ_{angle} The distance error is sensitive to the noisy false positive edges such as object surface textures. Therefore, we also detect line segments and measure whether they align with the computed vanishing points during cuboid generation in Sec 3.1. The angle alignment error is defined as:

$$\phi_{angle} = \|\theta(a, b) - \theta(p, b)\| \quad (2)$$

Where p is a VP and (a, b) is a detected line segment. θ is the line angle between two points.

Shape error ϕ_{shape} The previous two costs can be evaluated efficiently just in 2D image space. However, similar 2D cuboid corners might generate quite different 3D cuboids. We add a cost to penalize the cuboids with large skew ratio ($s = \text{length}/\text{width}$). More strict priors could be applied for example the estimated or fixed dimensions of vehicles, chairs and so on.

$$\phi_{shape} = \max(s - \sigma, 0) \quad (3)$$

where σ is a threshold. If $s < \sigma$, no penalty is applied.

4 Object SLAM

We extend the single image 3D object detection to multi-view object SLAM to further improve object detection accuracy and camera pose estimation.

We build our system on feature point based ORB SLAM [3], which includes the front-end of camera tracking and back-end of bundle adjustment. Our main change is the modified BA including objects, points and camera poses together, which will be explained detailedly in this section. Other SLAM implementation detail is in Section 5.1.

4.1 Bundle Adjustment Formulation

As we know, BA is the process to jointly optimize different map components including camera poses, points, lines and so on [3] [13]. Consider a set of camera poses by $C = \{c_i\}$ and 3D landmark object $O = \{o_j\}$. Points $P = \{p_k\}$ are also used in most of our experiments because objects alone usually cannot fully constrain camera poses. Then BA can be formulated as nonlinear least squares problem:

$$C^*, O^*, P^* = \arg \min_{\{C, O, P\}} \sum_{i \in C, j \in O, k \in P} \mathbf{e}^T W \mathbf{e} \quad (4)$$

where \mathbf{e} is the measurement error between C, O, P . W is the weight matrix for different errors. Definition of variables and errors are in the following. Then the optimization problem can be solved by many existing libraries such as g2o and iSAM.

Parameterization Camera poses are represented by $T_c \in SE(3)$ and points are represented by $P \in \mathbb{R}^3$. As explained in Section 3.1, cuboid objects are modelled as 9 DoF parameters: $O = \{T_o, D\}$ where $T_o \in SE(3)$ is 6 DoF pose, and $D \in \mathbb{R}^3$ is cuboid dimension.

4.2 Measurement Errors

Object-camera measurement We propose two kinds of measurement errors between objects and cameras depending on the scenarios.

The first kind is 3D measurement used when the 3D object detection is accurate. The detected object pose $O_m = (T_{om}, D_m)$ from Section 3.1 serves as the object measurement from the camera frame. Subscript m indicates measurement throughout the paper. To compute its measurement error, we transform the landmark object to the camera frame then compare with the measurement shown as:

$$e_{3D} = \| \log((T_c^{-1} T_o) T_{om}^{-1})_{\mathfrak{se}_3}^\vee \| + \| D - D_m \|_2 \quad (5)$$

where \log maps the $SE3$ error into tangent vector space suitable for optimization. Huber robust cost function is applied to all measurement error to improve the robustness [3].

We need to note that, without prior object model, our image-based cuboid detection cannot differentiate between the front or back of objects. For example, we can represent the same cuboid by rotating the object coordinate frame by

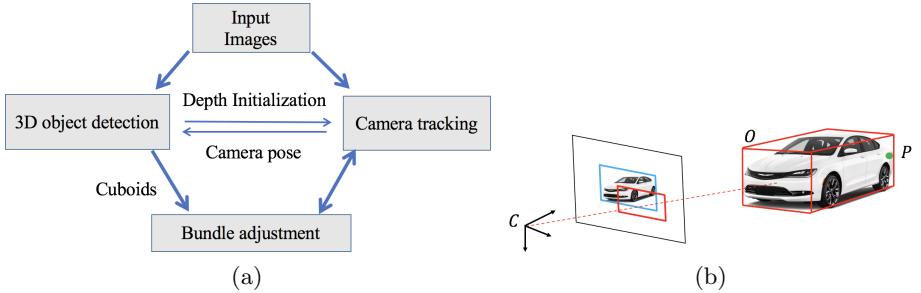


Fig. 4. (a) Our object SLAM pipeline. Single view object detection provides cuboid landmark and depth initialization for SLAM while SLAM can estimate camera pose for more accurate object detection. (b) Measurement errors between cameras, objects and points during BA.

90° and swapping length with width. Therefore, we need to rotate along height direction for ±90°, 0, 180° to find the smallest error in Eq 5.

The second kind is 2D measurement. We can project the landmark cuboid onto image plane to get the 2D bounding box shown as the red rectangles in Fig. 4(b) then compare it with the blue detected 2D bounding box:

$$e_{2D} = \| (c, d) - (c_m, d_m) \|_2 \quad (6)$$

where (c, d) is the center and dimension of 2D box. This measurement error has much less uncertainty compared to 3D error in Eq 5 because 2D object detection is usually more accurate compared to 3D detection which is similar to projecting map points onto images. But it also loses information after projection because many different 3D cuboids can project to the same 2D rectangle thus only one observation cannot fully constrain the camera poses and cuboids.

Estimating the object measurement uncertainty, or weight matrix W is not straightforward compared to point. We simply give more weights to confident and nearby objects. Suppose cuboid-camera distance is d and the object 2D object detection probability is p , then weight is $w = p \times \min((70 - d), 0)/50$ on KITTI outdoor data. Thresholds may vary with different scenarios.

Object-point measurement Points and objects can provide constraints for each other. If points belong to an object shown in Fig. 4(b), it should lie inside the 3D cuboid. Thus we can first transform the point to the cuboid frame then compare with cuboid dimensions:

$$e = \max(|(T_o^{-1} P)| - D_m, 0) \quad (7)$$

where max operator is used because we only encourage points to lie inside cuboid instead of on surfaces.

Point-camera measurement We use the standard 3D point re-projection error in feature based SLAM [3].



Fig. 5. Object association in dynamic and occluded scenarios in KITTI 07. Green points are normal map points, and other color points are associated to objects with the same color. The front cyan moving car is not added as SLAM landmark as no feature point is associated with it. Points in object overlapping areas are not associated with any object due to ambiguity.

4.3 Data association

Data association across frames is another important part. Compared to points, object association seems to be easier as more information contained and many 2D object tracking or template matching approaches can be used. Even simple 2D box overlapping can work in some simple scenarios. However, these approaches are not robust if there is severe object occlusion with repeated objects as shown in Fig. 5. In addition, dynamic objects need to be detected and removed from SLAM optimization but standard object tracking approaches cannot classify whether it is static or not, unless specific motion segmentation is used.

We thus propose another method for object association based on point matching. For many point based SLAM [3], dynamic points can be effectively detected through descriptor matching and epipolar line checking. Thus we first associate points to objects if points are observed enough times of belonging to the 2D object bounding box and close to cuboid centroid in 3D space. Some latest instance segmentation [1] could also be used to improve the point-object association accuracy. Then we can find object matching which has the most number of shareable map points exceeding a threshold (10 in our implementation). Note that this object-point association is also used when computing object-point measurement error during BA in Sec 4.2. Through our experiments, this approach works well for wide baseline matching, repetitive objects, occlusions, and dynamic scenarios explained in Fig. 5.

5 Experiments

5.1 Implementation

Object detection For the 2D object detection, YOLO detector [26] with probability threshold of 0.25 is used for indoor scenarios and MS-CNN [27] with probability of 0.5 is used for outdoor KITTI. Both can run in real time on GPU.

If accurate camera pose is known for example in SUN RGBD dataset, we only need to sample object yaw to compute VPs as explained in Section 3.1. 15 samples of object yaw in a range of 90° are generated as cuboids can be rotated

mentioned in Section 4.2. Then 10 points are sampled on the top edge of 2D bounding box. Note that not all the samples can form valid cuboid proposals as some cuboid corners might lie outside of the 2D box. In scenarios with no ground truth camera pose provided, we sample camera roll/pitch as well in the feasible range or based on SLAM pose estimation. One advantage of our approach is that it doesn’t require large training data as we only need to tune the two cost weights in Eq 1. It can also run in real time including 2D object detection and edge detection.

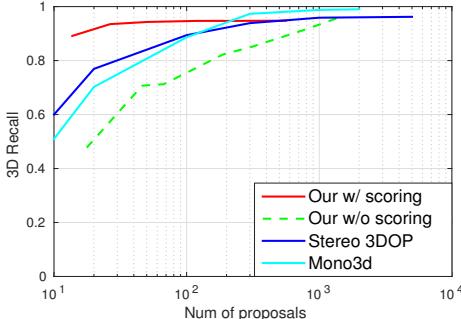
Object SLAM The pipeline of the whole SLAM algorithm is shown in Fig. 4(a). As mentioned in Sec 4, our system is based on ORB SLAM2 and we didn’t change camera tracking and keyframe creation module. For the newly created keyframe, we detect the cuboid objects, associate them, then perform bundle adjustment with camera poses and points. The cuboid is also used to initialize depth for feature points difficult to triangulate when stereo baseline or parallax is smaller than a threshold. This can improve the robustness in some challenging scenarios such as large camera rotations demonstrated in the experiments. Since the number of objects is far less compared to points, object association and BA optimization is very efficient in real time. To get an absolute map scale for monocular SLAM, the initial frame camera height is provided to scale the map.

Note that our object SLAM can also work independently without points. In some demonstrated challenging environments with few feature points, ORB SLAM cannot work, but our algorithm can still estimate camera poses using only object-camera measurement.

5.2 Single View 3D object detection

SUN RGBD [28] and KITTI [29] dataset with ground truth 3D bounding box annotations are used for single view object detection evaluation. 3D intersection over union (IoU) is adopted as the evaluation metric instead of only rotation or viewpoint evaluation in many other works. If the 3D IoU is greater than 25%, it is treated as positive detection [28] [11]. Since our approach doesn’t depend on the prior object model, in order to get an absolute scale of object position and dimensions, we only evaluate the ground objects with known camera height as explained in Sec 3.1. For the KITTI dataset, this assumption is already satisfied. For the SUN RGBD dataset, we select 1670 images with visible ground plane and ground objects fully in field of view.

Proposal Recall We first evaluate the proposal quality on the KITTI dataset using the training and validation split by [30]. Note that [11] first exhaustively samples huge amounts cuboid proposals ($\sim 14k$) then reports the recall after selecting the top N proposals based on semantic segmentation and so on. As the red line in Fig. 6(a), before scoring, our approach can reach a recall of 90% with 800 raw proposals per image, about 200 proposals per object. After scoring (green line), we can reach the same recall using just 20 proposals, much less compared to [11]. There are two main reasons for that: one is that our 3D proposals are of



(a)

	Method	3D IoU	Detected cuboids
SUN RGBD	Primitive[9]	0.36	125
	3dgp[31]	0.42	221
	Ours	0.39	1890
	Ours *	0.45	252
KITTI	Deep[12]	0.33	10957
	SubCNN[30]	0.21	8730
	Ours	0.21	10406
	Ours top 10	0.38	10406

* On 3dgp detected images.

(b)

Fig. 6. Single image 3D object detection evaluation (a) Proposal recall on KITTI dataset. Our approach can get high recall with low samples. (b) Comparison on SUN RGBD subset and KITTI dataset.

high quality because they are guaranteed to match the 2D detected box. Another reason is the more effective scoring function. Note our approach has an upper limit shown Fig. 6(a) because 2D detector might miss some objects.

Final detection We then evaluate the final accuracy of the best selected proposal. In SUN RGBD, to our knowledge, we didn't find the trained 3D detection algorithm on it. We thus compare with two public approaches SUN primitive [9] and 3D Geometric Phrases (3dgp) [31] for indoor environments. We modify their code to use the actual camera pose and calibration matrix when detecting and unprojecting to 3D space. To eliminate the effect of 2D detector, for all the methods, we only evaluate 3D IoU for objects with 2D box IoU >0.7 . As shown in Fig 6(b) and Fig. 7, our approach is more robust as it can detect many more accurate cuboids. The mean 3D IoU is smaller compared 3dgp using prior CAD models but higher if only evaluated on the same detected objects by 3dgp.

On the KITTI dataset, we compare with two other monocular algorithms [12] [30] using deep networks. From Fig 6(b), our approach performs similarly to SubCNN which uses prior object models. It performs worse compared to [12] which directly predicts vehicle orientations and dimension. As there is only one object class "car" with fixed camera viewpoints, CNN prediction works better than our hand-designed features. In the last row, we also include the evaluation of our selected top 10 cuboid proposals. It indicates that our proposals are still of high quality within only a few proposals.

5.3 Object SLAM

We then evaluate the performance of object SLAM, including camera pose estimation, and 3D object IoU after BA optimization. We show that SLAM and object detection can benefit each other in various datasets. Root mean squared error (RMSE) [32] and KITTI relative translation error [29] are used to evaluate the camera pose.



Fig. 7. 3D object detection examples in SUN RGBD and KITTI dataset (Single view).

TUM and ICL-NUIM RGBD dataset These datasets [32] [33] provide ground truth camera pose estimation and only RGB image is used here. We register a global point cloud using depth image and manually label 3D cuboids as the ground truth object.

We first test on TUM *fr3_cabinet* shown in Fig. 8(a) which is a challenging low texture dataset and existing monocular SLAM algorithms all fail on it due to few point features. We utilize object as the only SLAM landmark with 3D object-camera measurement in Sec 4.2. The left of Fig. 8(a) shows our online detected cuboid in some frames using estimated camera pose from SLAM. There is clearly large detection error in the bottom image. After multi-view optimization, the red cube in the map almost matches with the ground truth point cloud. From row "fr3/cabinet" in Fig. 8(b), 3D object IoU is improved from 0.46 to 0.64 after SLAM optimization compared to the mean single image cuboid detection. The absolute camera pose error is 0.17m. We then test on ICL living room dataset which is a general feature rich scenario. Since there is no absolute scale for monocular DSO or ORB SLAM, we compute the pose error after scale alignment [13]. We improve the object detection accuracy while sacrificing a bit camera pose accuracy. As can be seen from the object mapping of ICL data in Fig. 1(a), our approach is able to detect different objects including sofas, chairs, and pot-plant demonstrating the advantage of 3D detection without prior models.

Collected chair dataset We collect two chair datasets using Kinect RGBD camera shown in Fig. 9(a) 9(b). The RGBD ORB SLAM result is used as the ground truth camera poses. The second dataset Fig. 9(b) contains large camera rotation which is challenging for most monocular SLAM. As shown in Fig. 9(a), after optimization, cuboids can fit tightly with the associated 3D points. The quantitative error is shown in the bottom two rows of Fig 8(b). DSO is able to work in the first dataset but performs very bad in the second one due to

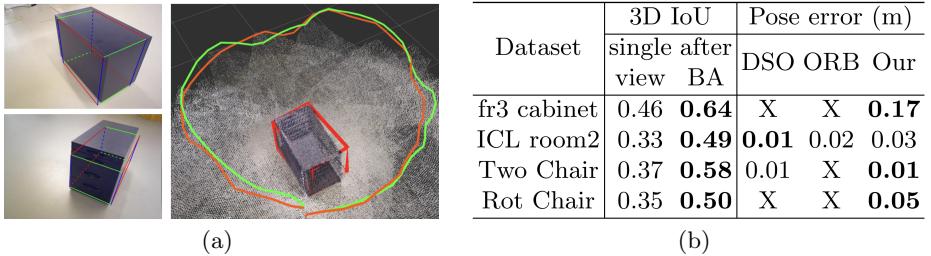


Fig. 8. (a) Object SLAM on TUM fr3_cabinet. Red cube is the optimized object landmark, matching well with the truth point cloud. Red and green trajectories are the predicted and truth camera paths. Existing SLAMs fail on this dataset due to low texture. (b) Quantitative result of object detection and pose estimation on various indoor datasets.



Fig. 9. Collected chair datasets. (a) Objects fit tightly with the associated points. (b) Objects improve camera pose estimation when there is large camera rotation.

large camera rotation. Mono ORB SLAM fails to initialize in both cases while our cuboid detection can provide depth initialization for points even from single image. Similar as before, the 3D object IoU is also improved after BA.

KITTI Dataset We test on two kinds of KITTI dataset, one is short sequence with provided ground truth object annotations, the other is long sequence for standard odometry benchmark without object annotations. We use the 2D object-camera measurement in Sec 4.2 during BA because of low uncertainty. When we compare with ORB SLAM, we turn off its loop closure module in order to better measure the monocular scale drift. We also scale ORB SLAM’s initial map by the first frame camera height (1.7m in our implementations), then we can directly evaluate the absolute trajectory error without aligning the trajectory in scale. In Fig 10, we can observe that the initial trajectory segment before first turning matches well with ground truth, indicating the initial map scaling is correct. For KITTI dataset, we additionally initialize object dimension using prior car size ($w = 3.9, l = 1.6, h = 1.5$ in our implementation) to maintain long-term scale consistency, which is also used in other object SLAM works [19][20]. This is especially useful when objects are not observed frequently in some sequence.

For the first category data, we select 10 kitti raw sequences with the most number of ground truth object annotations named ‘2011_09_26_drive_xx’. As shown in Table 1, both 3D object IoU and camera pose accuracy are improved. Especially for camera pose, object SLAM can provide geometry constraints to

Table 1. Object Detection and Camera Pose Estimation on KITTI Object Sequence

	Seq	22	23	36	39	61	64	93	95	96	117	Mean
Object 3D IoU	Single view[12]	0.52	0.32	0.50	0.54	0.54	0.43	0.43	0.40	0.26	0.25	0.42
	Ours	0.58	0.35	0.54	0.59	0.50	0.48	0.45	0.52	0.29	0.35	0.47
Trans error(%)	ORB SLAM	13.0	1.17	7.08	6.76	1.06	7.07	4.40	0.86	3.96	4.10	4.95
	Ours	1.68	1.72	2.93	1.61	1.24	0.93	0.60	1.49	1.81	2.21	1.62

Table 2. Camera Pose Estimation Error on KITTI Odometry Benchmark

	Seq	0	2	3	4	5	6	7	8	9	10	Mean	
Trans Error (%)	Ground based	[34]	4.42	4.77	8.49	6.21	5.44	6.51	6.23	8.23	9.08	9.11	6.86
	[35]	2.04	1.50	3.37	1.43	2.19	2.09	-	2.37	1.76	2.12	-	2.03
Error (%)	Object based	[20]	3.09	6.18	3.39	32.9	4.47	12.5	2.81	4.11	11.2	16.8	9.75
	Ours	2.40	4.25	2.87	1.12	1.64	3.20	1.63	2.79	3.16	4.34	2.74	
Combined	Ours	1.97	2.48	1.62	1.12	1.64	2.26	1.63	2.05	1.66	1.46	1.78	
RMSE (m)	Object based	[19]	73.4	55.5	30.6	10.7	50.8	73.1	47.1	72.2	31.2	53.5	49.8
	Ours		13.9	26.2	3.79	1.10	4.75	6.98	2.67	10.7	10.7	8.37	8.91

reduce the scale drift of monocular SLAM. Visualization of some object mapping and pose estimation are shown in Fig 1(b) and Fig 10, where we can see our approach greatly reduces monocular scale drift.

For the KITTI odometry dataset, most existing monocular SLAM use constant ground plane height assumption to reduce monocular scale drift [34] [35]. Recently, there are also some object based scale recovery approaches [20] [19]. Result of them is taken from their paper directly. We didn't compare with ORB SLAM in this case as without loop closure, it cannot recover scale in the long sequence and has significant drift shown in Fig 10. As shown in Table 2, our object SLAM performs much better compared to other SLAM using objects, because they represent vehicles as spheres or only utilize vehicle height information, which is not accurate compared to our cuboid representation. Our algorithm is also comparable to ground-based scaling approaches. It performs worse in some sequences such as Seq 02, 06, 10, mainly because there are not many objects visible over long distance thus causing scale drift. Therefore, we also propose a simple method to combine ground height assumption with object SLAM. If there is no object visible in recent 20 frames, we do point cloud plane fitting then scale camera poses and local map using the constant ground height assumption. As shown in row 'Combined' in Table 2, it achieves the state-of-art accuracy of monocular SLAM on KITTI benchmark. Note that ground plane based approaches also have their limitations for example not applicable for aerial vehicle or handheld camera. It will also fail if ground is not visible such as frames in Fig 5 of KITTI 07. The front dynamic car occludes the ground for a long time and that's why many ground-based approaches fail or perform poorly on KITTI 07.

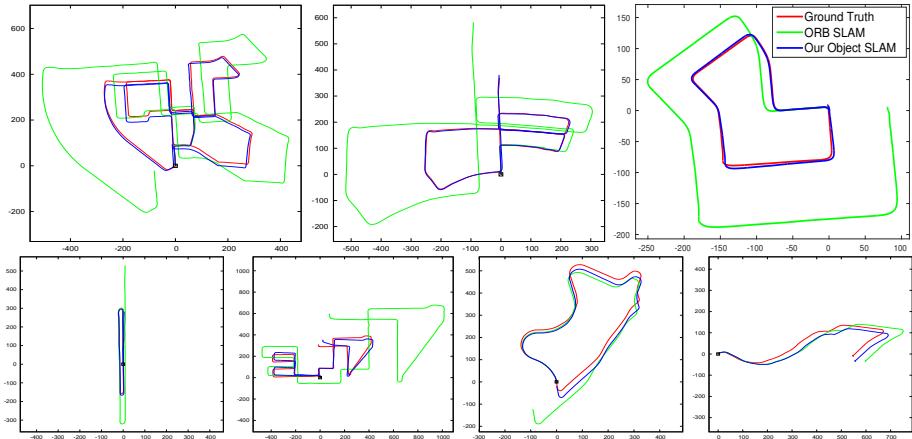


Fig. 10. Our object SLAM on KITTI odometry dataset without loop closure and constant ground height assumption. Red is ground truth. Blue is our object SLAM. Green is ORB point SLAM. Objects can reduce monocular scale drift and improve pose estimation accuracy.

6 Conclusion

In this work, we propose a general approach for monocular 3D object detection and SLAM mapping without prior object models. More importantly, we demonstrate for the first time, that semantic object detection and geometric SLAM can benefit each other in one unified framework.

For the single image 3D object detection, we propose a new method to efficiently generate high quality cuboid proposals from 2D bounding box based on vanishing points. Proposals are then scored efficiently by image cues. In the SLAM part, we propose an object level SLAM with novel measurement functions between cameras, objects and points, and new object association methods to handle robustly with occlusion and dynamic movement. Objects can provide depth constraints for points and scale constraints for camera poses. In turn, SLAM also provides camera pose initialization for detecting and refining 3D object detection.

We evaluate the two parts on different indoor and outdoor datasets and achieve the best accuracy of 3D object detection on SUN RGBD subset data and camera pose estimation on KITTI odometry datasets. In the future, we are also interested in considering dynamic objects and dense mapping using objects. More complete scene understanding can be integrated with SLAM optimization.

References

1. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. IEEE International Conference on Computer Vision (ICCV) (2017)
2. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: IEEE Winter Conference on Applications of Computer Vision (WACV). (2014) 75–82
3. Mur-Artal, R., Montiel, J., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Transactions on Robotics (2015) 1147–1163
4. Yang, S., Scherer, S.: Direct monocular odometry using points and lines. In: International conference on Robotics and automation (ICRA), IEEE (2017)
5. Lim, J.J., Pirsiavash, H., Torralba, A.: Parsing ikea objects: Fine pose estimation. In: IEEE International Conference on Computer Vision(ICCV). (2013) 2992–2999
6. Bansal, A., Russell, B., Gupta, A.: Marr revisited: 2d-3d alignment via surface normal prediction. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5965–5974
7. Chabot, F., Chaouch, M., Rabarisoa, J., Teuli  re, C., Chateau, T.: Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR). (2017)
8. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: IEEE International Conference on Computer Vision (ICCV). (2017)
9. Xiao, J., Russell, B., Torralba, A.: Localizing 3d cuboids in single-view images. In: Advances in neural information processing systems. (2012) 746–754
10. Hedau, V., Hoiem, D., Forsyth, D.: Thinking inside the box: Using appearance models and context based on room geometry. In: European Conference on Computer Vision, Springer (2010) 224–237
11. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2147–2156
12. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3D bounding box estimation using deep learning and geometry. Computer Vision and Pattern Recognition (CVPR), IEEE Conference on (2017)
13. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
14. Dong, J., Fei, X., Soatto, S.: Visual-inertial-semantic scene representation for 3d object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
15. S  nderhauf, N., Pham, T.T., Latif, Y., Milford, M., Reid, I.: Meaningful maps with object-oriented semantic mapping. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE (2017) 5079–5085
16. Pillai, S., Leonard, J.: Monocular slam supported object recognition. Robotics: Science and systems (2015)
17. Bao, S.Y., Bagra, M., Chao, Y.W., Savarese, S.: Semantic structure from motion with points, regions, and objects. In: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2012) 2703–2710
18. Salas-Moreno, R.F., Newcombe, R.A., Strasdat, H., Kelly, P.H., Davison, A.J.: Slam++: Simultaneous localisation and mapping at the level of objects. In: IEEE Conference on Computer Vision and Pattern Recognition. (2013) 1352–1359

19. Frost, D.P., Kähler, O., Murray, D.W.: Object-aware bundle adjustment for correcting monocular scale drift. In: International Conference on Robotics and Automation (ICRA), IEEE (2016) 4770–4776
20. Sucar, E., Hayet, J.B.: Bayesian scale estimation for monocular slam based on generic object detection for correcting scale drift. In: International Conference on Robotics and Automation (ICRA), IEEE (2018)
21. Gálvez-López, D., Salas, M., Tardós, J.D., Montiel, J.: Real-time monocular object slam. *Robotics and Autonomous Systems* **75** (2016) 435–449
22. Gay, P., Bansal, V., Rubino, C., Del Bue, A.: Probabilistic structure from motion with objects (psfmo). In: IEEE International Conference on Computer Vision (ICCV). (2017) 3075–3084
23. Mu, B., Liu, S.Y., Paull, L., Leonard, J., How, J.P.: Slam with objects using a nonparametric pose graph. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2016) 4602–4609
24. Bowman, S.L., Atanasov, N., Daniilidis, K., Pappas, G.J.: Probabilistic data association for semantic slam. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on, IEEE (2017) 1722–1729
25. Yang, S., Song, Y., Kaess, M., Scherer, S.: Pop-up SLAM: a semantic monocular plane slam for low-texture environments. In: IEEE international conference on Intelligent Robots and Systems (IROS). (2016)
26. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. *Computer Vision and Pattern Recognition (CVPR)* (2017)
27. Cai, Z., Fan, Q., Feris, R., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: ECCV. (2016)
28. Song, S., Lichtenberg, S.P., Xiao, J.: Sun RGB-D: A RGB-D scene understanding benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 567–576
29. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2012) 3354–3361
30. Xiang, Y., Choi, W., Lin, Y., Savarese, S.: Subcategory-aware convolutional neural networks for object proposals and detection. In: Winter Conference on Applications of Computer Vision (WACV), IEEE (2017) 924–933
31. Choi, W., Chao, Y.W., Pantofaru, C., Savarese, S.: Understanding indoor scenes using 3d geometric phrases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 33–40
32. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: International Conference on Intelligent Robots and Systems (IROS), IEEE (2012) 573–580
33. Handa, A., Whelan, T., McDonald, J., Davison, A.: 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)
34. Lee, B., Daniilidis, K., Lee, D.D.: Online self-supervised monocular visual odometry for ground vehicles. In: IEEE International Conference on Robotics and Automation (ICRA), IEEE (2015) 5232–5238
35. Song, S., Chandraker, M., Guest, C.C.: High accuracy monocular sfm and scale correction for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence* **38**(4) (2016) 730–743