2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI) | 978-1-6654-4521-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/MFI52462.2021.9591174

Object Detection and Mapping with Bounding Box Constraints

Benchun Zhou[1], Aibo Wang[1], Jan-Felix Klein[1], Furmans Kai[1]

***Abstract*— In this paper, we present a three-dimensional object detection method for a single image and an object-based localization and mapping system. For 3D object detection, we firstly generate high-quality cuboid candidates by sampling object rotation and dimension. Then, the translation of each candidate is estimated in a closed form solution with camera projection function and bounding box constraints. Finally, all candidates are projected into the image, scored and selected based on the alignment with detected lines. To overcome object detection accuracy issues, the results are improved by multi-view optimization. Besides, objects can provide geometry constraints and semantic information to improve camera pose estimation and monocular drift. A point-object SLAM system is formulated to jointly optimize the poses of camera, objects and points. We evaluate our object detection method on objects from the KITTI, the SUN RGB-D and a self collected dataset. The results show that our method outperforms existing approaches. The point-cuboid SLAM experiments on the TUM RGB-D, ICL-NUIM and our self collected dataset show that our algorithm can improve both camera localization accuracy and 3D object detection accuracy.**

Keywords: 3D object detection, object based SLAM, monocular, bounding box constraints.

I. INTRODUCTION

In intralogistic environments, many objects, such as parcels and containers, are standard cuboids. These objects can not only provide valuable information for navigation and grasping, but are also beneficial for creating a meaningful map including geometry and object information. A sensor fusion system is commonly the best choice to create such a map. Taking advantages of LiDAR, camera, IMU and other sensors, this system can achieve state-of-art results. But on the other hand, the complexity and overall cost will be increased. In this paper, we focus on a monocular camera system, which is cheap and flexible. Visual localization and mapping systems (SLAM) with feature points achieved a significant success in indoor environments [1]. However, since no object information is included, the map does not provide information necessary for scene understanding. 3D object perception, on the other hand, can provide geometry and semantic information [2]. Currently, the mainstream 3D object detection methods use deep learning framework to predict object information, but require hundreds of labeled training data and can only be applied in specific environment [3]. To make the system universal, we focus on traditional image features and propose a general object detection method. When object information is available, it can be integrated into feature points based SLAM, and jointly optimizes camera localization, points pose and object information through bundle adjustment (BA).

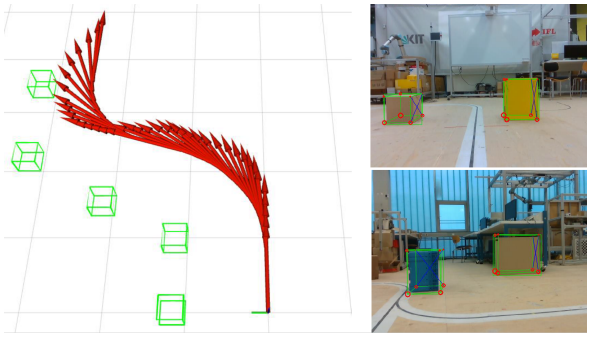

Fig. 1. Example result of object detection and vehicle trajectory on collected data

In this work, we propose a novel method to combine 3D object detection with SLAM pose estimation. Given a RGB image, the 2D object information can be easily obtained, based on that, many 3D cuboid proposals can be effectively generated through orientation and dimension sampling. Under the assumption that the perspective projection of a 3D cuboid should fit tightly with its 2D detection bounding box (bbox), the translation of each proposal can be estimated by non-linear least squares functions. After obtaining cuboid proposals in 3D space, we project all proposals onto the image plane, score them with image features and select the cuboid that fits the object best. This detected cuboid is further optimized with points and camera through a multi-view BA framework, to improve the pose of camera, objects and points. In summary, our contributions are as follows:

- An accurate, effective and robust single image 3D cuboid detection approach without training data.
- An object-based SLAM framework, including camera, object and point information, that improves pose estimation accuracy.
- Results on public and collected dataset demonstrate the effectiveness of our approach.

In the following, the related work is introduced in Section II, object detection method and object-based SLAM is explained in Section III and IV respectively. Then, the experiments and the results are shown in Section V and VI. Finally, Section VII concludes the paper. Part of the code is available at `https://github.com/benchun123/cuboid_slam_with_bbox_constraints.git`.

*Research supported by China Scholarship Council (CSC) Foundation.

[1]Benchun Zhou, Aibo Wang, Jan-Felix Klein, Kai Furmans are with the Institute for Material Handling and Logistics (IFL), Karlsruhe Institute of Technology (KIT), Karlsruhe, 76131, Germany `benchun.zhou@kit.edu, aibowang@foxmail.com, jan-felix.klein@kit.edu, kai.furmans@kit.edu`

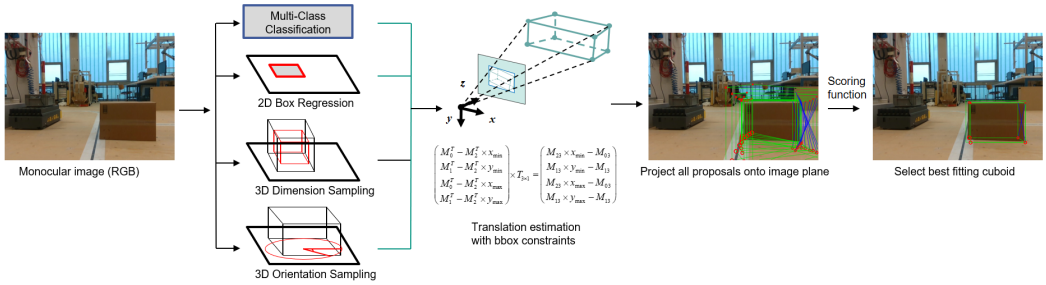

Fig. 2. Flow chart of single image 3D object detection, including sample, estimation and score stage

II. RELATED WORK

A. Single Image 3D Object Detection

Monocular 3D perception remains a very difficult challenge, because only 2D information is known. According to the representation models, object detection methods can be divided into two categories: with or without object models. When object models are known, they can be detected by matching the RGB image with model keypoints or features [4]. Without prior models, objects are usually represented by cuboids, and the parameters are estimated by traditional or deep learning approaches.

Traditional approaches take advantages of features, such as lines and planes to compute a reliable result [5], while deep learning approaches rely on training data and neural network to predict the object's dimension and rotation [6]. Since intralogistics environments are low-texture and contain many boxy like objects, we focus on traditional 3D object detection approaches with cuboid format. A similar approach is provided by CubeSLAM [7], which samples the 2D bounding box as cuboid corners, computes cuboid with geometry relationships of edges, scores them based on image lines and selects the best fit cuboid.

B. Object-based SLAM

Feature points based visual SLAM algorithms, such as ORB SLAM 2[8], take feature points as landmarks and can simultaneous locate the camera and map the environment. Similarly, object-based SLAM methods use objects as landmarks to achieve the same goal but add semantic information.

Different object models are used in SLAM framework. Salas [9] proposed a practical object SLAM system called SLAM++, using RGB-D cameras and prior object models. Siddharth [10] modeled objects as segmentation of a point cloud. Quadrics model can be computed by multi-view optimization in [11], and was extended to Quadrics SLAM [12]. Yang [7] proposed CubeSLAM, where the objects are presented as cuboids and the SLAM system successfully applied in both static and dynamic environments. Our system shares the same idea with CubeSLAM, but uses a different way to detect cuboid.

From a single image, objects have limited number and are much larger than feature points. They will cause big errors when integrated into SLAM framework. Therefore, a comprehensive system with feature points and objects is introduced to practical environment. Yang [13] proposed a monocular SLAM and dense mapping algorithm, combining points with high-level object and plane landmarks. Mehdi [14] also introduced a monocular SLAM that incorporated plane, points and quadrics in an online real-time capable system.

III. SINGLE IMAGE 3D OBJECT DETECTION

A general cuboid can be represented by 9 DoF parameters: 3 DoF position $\mathrm{t} = [\mathrm{t_x}, \mathrm{t_y}, \mathrm{t_z}]$, 3 DoF rotation R, and 3 DoF dimension $\mathrm{d} = [\mathrm{w}, \mathrm{h}, \mathrm{l}]$. Many deep learning methods use training data to predict them. Rather than relying on predicted orientation and dimension, we can directly sample them. Each dimension and orientation pair becomes a cuboid proposal. We assume that the cuboid's projected vertexes should fit the 2D bounding box tightly, then, the translation of the cuboid can be computed by a closed form solution. Then, we project all proposals onto the image plane, score them with geometry features, and select the best cuboid that has minimum cost. The overall process is shown in Fig 2.

A. Stage 1: Sample

There are various 2D object detection methods, for example YOLOv4 [15], which can classify objects and estimate 2D bounding boxes with high accuracy. We consider these detection results to be reliable and use them for the subsequent process. We also need to mention that the quality of the computed 2D bounding boxes have a great influence on the following 3D object detection.

Estimating the dimension of objects from a single image is a difficult task. Sampling the length, width and height from 0m to 10m takes much time and is not practical. Instead, objects from KITTI [16] and SUN RGB-D dataset [17] provide a lot of object ground truth information, which one can use to formulate an average dimension benchmark. We can therefore sample around the average dimension by object name.

For general objects, we also need to sample roll, pitch, yaw in every axis. However, objects are always lying on a supporting plane, such as a static ground, a desk or a wall. These planes are usually parallel or vertical to the world, which allows us to sample only a single axis that is perpendicular to the supporting plane. Therefore, the sampling space is greatly reduced and becomes more accurate. In this paper, we only consider objects that are parallel to the ground.

B. Stage 2: Estimate Translation with BBox Constraints

For a perspective camera, given the camera intrinsic matrix K, the 2D bounding box $[\mathrm{x_{min}};\mathrm{y_{min}};\mathrm{x_{max}};\mathrm{y_{max}}]$, the dimension $\mathrm{d}=[\mathrm{w},\mathrm{h},\mathrm{l}]$ and the orientation of the object $R(\theta)$, the projection function is:

$$\lambda\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathrm{P}_{roj}\begin{pmatrix} X_{3D} \\ 1 \end{pmatrix} = \mathrm{P}_{roj}\begin{pmatrix} I_{3\times3} & R_{3\times3}(\theta)\times D_{3\times1} \\ 0 & 1 \end{pmatrix}\begin{pmatrix} T_{3\times1} \\ 1 \end{pmatrix} \quad (1)$$

where (u,v) is the point in image, and the projection matrix P_{roj} can be a camera intrinsic matrix K when X_{3D} is in camera coordinates. Then, X_{3D} can be decomposed by object translation T, orientation $R(\theta)$ and dimension D.

Assuming that the cuboid's projected corners should fit the 2D bounding box tightly, four equations from left x_{min}, right x_{max}, top y_{min} and bottom y_{max} constraints can be obtained. Take the right constraints as an example:

$$x_{\max} = \left(\begin{pmatrix} M[0,0:3] & M[0,3] \\ M[1,0:3] & M[1,3] \\ M[2,0:3] & M[2,3] \end{pmatrix}\begin{pmatrix} T_{3\times1} \\ 1 \end{pmatrix}\right)_x \quad (2)$$

$$\left(M_0^T - M_2^T \times x_{\max}\right)\times T_{3\times1} = M_{23}\times x_{\max} - M_{03} \quad (3)$$

Where $M = P_{roj}\times(I|R\times D)$, $M_0^T = M[0,0:3]$, $M_{03} = M[0,3]$, Similarly, if we take four 2D bbox constraints into consideration, we can get an over-constrained system as follows, which can be solved by the least squares method:

$$\begin{pmatrix} M_0^T - M_2^T\times x_{\min} \\ M_1^T - M_2^T\times y_{\min} \\ M_0^T - M_2^T\times x_{\max} \\ M_1^T - M_2^T\times y_{\max} \end{pmatrix}\times T_{3\times1} = \begin{pmatrix} M_{23}\times x_{\min} - M_{03} \\ M_{13}\times y_{\min} - M_{13} \\ M_{23}\times x_{\max} - M_{03} \\ M_{13}\times y_{\max} - M_{13} \end{pmatrix} \quad (4)$$

$$A\times T_{3\times1} = b(b\neq0) \rightarrow T_{3\times1} = \left(A^TA\right)^{-1}A^Tb \quad (5)$$

The final question is how to choose the four points out of eight of the 3D cuboid corners to be projected on the four sides of 2D bbox respectively. Assuming that objects lay on the ground plane and that by fixing one vertical 3D edge the second vertical one must be diagonally opposed, we have $4\times1\times4\times4=64$ configurations - from which we choose the best fit [20].

C. Stage 3: Score

After estimating the translation, we project the proposals into image, and define a cost function to score them. There are many score functions, such as semantic segmentation [21] or HoG features [22]. Following [7], we use the fast and effective cost functions to align the cuboid with edge features, while in intralogistics environment, most objects are "boxy with clear edge". The cost function between image I and object O is defined as:

$$E(O\mid I) = \omega_1\phi_{\mathrm{bbox}}(O\mid I) + \omega_2\phi_{\mathrm{angle}}(O\mid I) + \omega_3\phi_{\mathrm{dist}}(O\mid I) \quad (6)$$

where $\phi_{\mathrm{bbox}}(O\mid I)$, $\phi_{\mathrm{angle}}(O\mid I)$, $\phi_{\mathrm{dist}}(O\mid I)$ are bbox costs, edge angle alignment costs and distance costs, and ω_{123} are the weights, which need to be manually set for different datasets.

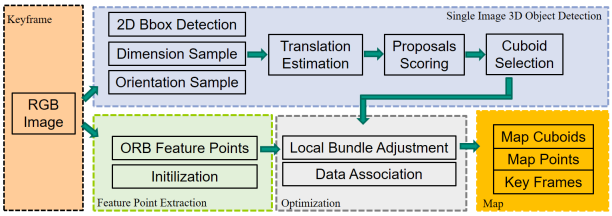

Fig. 3. Overall concept of point-object SLAM framework

1) 2D bounding box cost: The bounding box cost measures the fit of the cuboid's projected bounding box with the detected one, which is an effective way to eliminate proposals that have a large error. the cost can be computed by the projected bbox with center and size $[c_{\mathrm{proj}}, s_{\mathrm{proj}}]$ and detected bbox $[c_{\mathrm{det}}, s_{\mathrm{det}}]$:

$$\phi_{bbox}(O\mid I) = \left\|[c_{proj}, s_{proj}] - [c_{det}, s_{det}]\right\|^2 \quad (7)$$

2) Angle alignment cost: The angle alignment cost measures the angle between the cuboid's projected edges and the detected lines in the image. We first detect long line segments, simply filter some lines that are short or outside the 2D bounding box, then, match them to the cuboid's projected edges by minimum distance. We denote the detected line angle as θ_i, and the cuboid's edge as φ_k. Then, the angle alignment cost is calculated as follows, where k is the edge index:

$$\phi_{\mathrm{angle}}(O\mid I) = \sum_{i=1:n}\|\theta_i - \varphi_k\|, k\in[1,12], n<12 \quad (8)$$

3) Distance Cost: The distance cost is mainly used to measure how close the cuboid's projected 2D edges correspond with the ones inside the image. There are two ways. First, we can compute a distance transform map based on canny edges, and measure the projected edge value inside the distance map. Second, similar to the angle alignment cost, we match the lines and edges l_j, sample 10 points on each line and compute the distance from the sample points p_i to the cuboid's edges. The distance cost is then computed as:

$$\phi_{\mathrm{dist}}(O\mid I) = \frac{1}{n}\sum_{i=1:n}\sum_{j=1:m}\mathrm{dist}(p_i, l_j)\|, n<12, m<12 \quad (9)$$

The best proposal with the minimum cost is selected and can be transformed from camera coordinates into world coordinates when camera pose is available.

IV. OBJECT BASED LOCALIZATION AND MAPPING

We further extend the proposed 3D object detection from a single image to a multi-view object SLAM which jointly optimizes object pose and camera pose. To make the system more accurate and robust, we build object-based SLAM on ORB SLAM 2 [8] and introduce the camera-object error as a part of bundle adjustment formulation.

A. BA Formulation

According to [7], bundle Adjustment is the process to jointly optimize the camera pose $C \in SE(3)$, feature points $P \in \mathbb{R}^3$ and the cuboid $O = [T_o, d]$ where $T_o = [R, d] \in SE(3)$ is the 6 DoF pose, and $d \in \mathbb{R}^3$is the cuboid dimension. It can be formulated as a nonlinear least squares problem:

$$C^*, O^*, P^* = \underset{\{C,O,P\}}{\arg\min} \sum_{C_i,O_j,P_k} \left\| e(c_i, o_j) \right\|^2_{\Sigma_{ij}} + \left\| e(c_i, p_k) \right\|^2_{\Sigma_{ik}} + \left\| e(o_j, p_k) \right\|^2_{\Sigma_{jk}} \tag{10}$$

where $e(c_i, o_j)$, $e(c_i, p_k)$ and $e(o_j, p_k)$ represent the measurement errors between camera and objects and camera and points, and objects and points. Σ is co-variance matrix of different error measurements. Definitions of variables and errors are explained in the following.

B. Measurement Errors

1) Camera-Point Measurement: We follow ORB SLAM 2 [8] to minimize the reprojection error between matched 3D points P in world coordinates and keypoints z_m :

$$e(c_i, o_j) = \pi\left(T_c^{-1} P\right) - z_m \tag{11}$$

where π is the projection matrix and $T_c = [R, d]$ is the camera transformation matrix.

2) Camera-Object Measurement: Similar to camera-point error, we project cuboid landmarks O onto the image plane to get 8 corners and find the minimum distance between the landmark corners and detected corners y_m:

$$e(c_i, o_j) = \pi\left(T_c^{-1} O\right) - y_m \tag{12}$$

where π and $T_c = [R, d]$ are defined ahead.

3) Object-Point Measurement: Following CubeSLAM [7], if a point P belongs to an object, it should lie inside the 3D cuboid. we first transform the point to the cuboid frame, then compare with cuboid dimensions to get 3-D error:

$$e(o_j, p_k) = \max\left(\left|T_o^{-1} P\right| - \mathbf{d}_m, \mathbf{0}\right) \tag{13}$$

where *max* operator is used because we only encourage points to lie inside cuboid instead of exactly on surfaces. Huber robust cost function is applied to all measurement errors to improve the robustness [8].

C. Data Association

Data association across frames for different landmarks is an important part of SLAM. For feature points association, we followed ORB SLAM2 [8] to use point feature matching. For point object association, we consider the relative position between points and 2D bbox, and object association is solved by finding the most number of shared feature points that belong to the object [7].

V. EXPERIMENTS: SINGLE VIEW DETECTION

We evaluate our single image object detection method on KITTI object [16] and SUN RGB-D [17] dataset with ground truth 3D bounding box annotations. 3D intersection over union (IoU) is adopted as the evaluation metric. Since CubeSLAM [7] is similar to our presented approach, we follow the paper and compare them together. For KITTI object dataset, 1008 images with cars are tested, for SUN RGBD dataset, we select 200 images with different objects. The yaw angle is sampled from 0 to 180 degree every 5 degree, dimension from average to -+0.3m every 0.1m.

TABLE I
COMPARISON OF OBJECT 3D IOU RESULT

	KITTI	SUN RGBD	Logistic Dataset
Primitive[22]	–	0.36	–
3DGP[23]	–	**0.42**	–
Deep3dBox[6]	**0.33**	–	–
Mono3D[21]	0.22	–	–
CubeSLAM[7]	0.21	0.39	0.32
Ours*	**0.33**	0.31	**0.41**

– means no results found in reference

Fig. 4. Example results on KITTI, SUN RGB-D and collected dataset

A. KITTI Object Dataset and SUN RGB-D Datase

We share the similar idea with some deep learning methods, such as Deep3dBox [6] and Mono3D [21], so, we also evaluate our method on the KITTI object dataset [16] and mainly focus on car class. Instead of using neural networks to predict dimension and orientation of the cars, we directly sample them and use image features to score them. Table I shows that the detection results of our method is comparable to Deep3dBox, and outperform CubeSLAM detection method.

For indoor object detection in SUN RGB-D dataset [17], our method is hard to get a robust result due to object occlusion and messy background.

B. Logistic Object Dataset

We also evaluated our method in an intralogistics environment, which includes some parcels and containers. A Realsense camera D435 was used for data acquisition. We assume camera pose as the coordinates origin and get ground truth from point cloud. Compared to CubeSLAM detection, our method will get nearly 300 proposals and can get better estimation results. Besides, the scale of objects in CubeSLAM relies on the camera height, while our method relies on the 2D bbox and dimension.

TABLE II

OBJECT 3D IOU IN TUM CABINET DATASET

	Before Optimization	After Optimization
CubeSLAM (only object)	0.46	0.64
Ours (only object)	0.56	**0.75**

TABLE III

RESULT OF TRANSLATION ABSOLUTE TRAJECTORY ERROR

	TUM Cabinet	ICL room 2	IFL Line	IFL S-line
CubeSLAM (orb+object)	0.17	**0.03**	**0.05**	0.09
Ours (only object)	**0.08**	0.17	0.15	0.21
Ours (orb+object)	–	**0.03**	**0.05**	**0.08**

C. Analysis

Our object detection method can be applied in various scenes and works best for large and boxy like objects. Sampling and scoring are important parts of the object detection workflow. The benchmark of object dimension comes from a collected data, which is not convenient, and the scoring function should be further explored for different objects. Besides, the 2D bbox also influence translation estimation, if the 2D bbox becomes smaller, the estimated distance between camera with object will be larger.

VI. EXPERIMENTS: OBJECT SLAM

The performance of object SLAM is evaluated on the TUM Cabinet [18], the ICL-NUIM [19] and our collected intralogistic dataset. The root mean squared error (RMSE) is used to measure the camera pose and the 3D IoU between optimized objects and point cloud is considered to measure the object pose. For comparison, we implemented multi-object SLAM without feature points, orb-object SLAM and CubeSLAM.

A. TUM RGB-D Dataset and ICL-NUIM Dataset

The TUM Cabinet dataset contains a single object on the ground. Traditional point feature-based SLAM algorithms fail on this dataset due to the low texture, while object-based SLAM can output reliable results. As shown in table II, the average object 3D IoU from a single image reaches 0.56 and is improved to 0.75 after multi-view SLAM optimization. Both results exceed CubeSLAM. The results for the absolute camera pose error is displayed in table III. Our method is 0.11m, smaller than CubeSLAM. Figure 5 shows the detection and SLAM results of our method in comparison to CubeSLAM.

We also evaluate point-object SLAM in ICL NUIM dataset. The camera translation absolute errors after scale alignment are presented in Table III. SLAM with only object

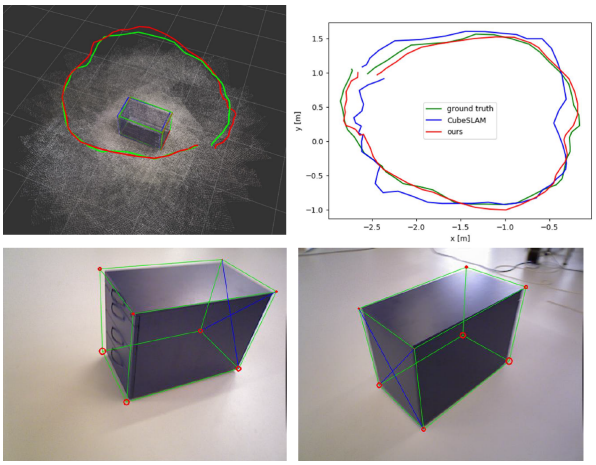

Fig. 5. Object SLAM result on TUM cabinet dataset

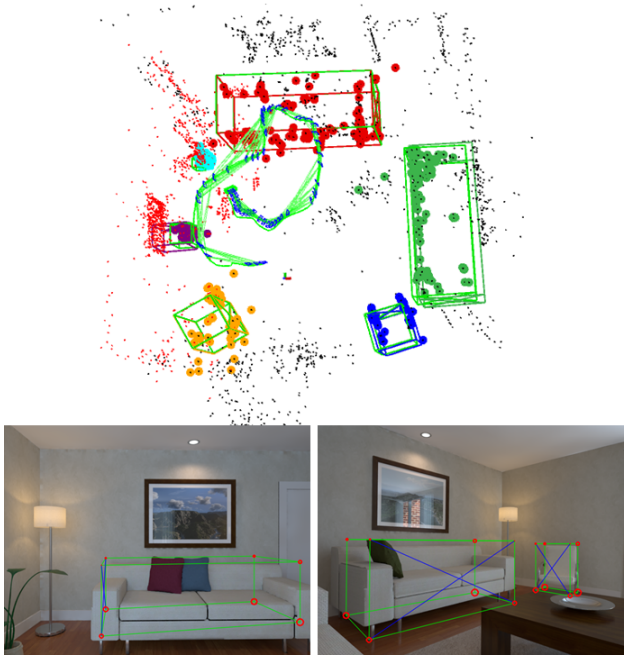

Fig. 6. Point-object SLAM result on ICL-NUIM dataset

is not stable and has the biggest error. our method is slightly better than CubeSLAM because of better object detection. A visualization of the result is shown in Fig 6, together with single image detection examples.

B. Logistic SLAM Dataset

To collect SLAM dataset on logistic environment, we build a capture system on a mobile vehicle and use a joystick to control its motion. We think the vehicle odometry is good and compute the camera truth pose by transforming the odometry from odom link to camera link. The start point of vehicle is set as world coordinates origin, and measure the object pose relative to the start points. Realsense camera D435 is used to collect RGB images with frequency 30 fps. The camera is parallel to the vehicle but heading to the left, so that the camera can move laterally and ORB SLAM 2[8] can

be initialized. For the intralogistics environment, the vehicle follows a line and a "S" line, the trajectory of vehicles can be found in Fig 1.

The camera translation absolute errors are shown in table III, from which we can see our multi-object SLAM framework work poorly and is not practical for real environment. CubeSLAM and our method are comparable, both can build an object-based map with different objects and feature points.

C. Analysis

We build our system on ORB SLAM and object detection, the object can provide geometry constraints and semantic information to the system, but the result heavily relies on single image object detection. Inaccurate object detection may lead to significant errors. Besides, the scale of the map is difficult for monocular SLAM system.

VII. CONCLUSIONS

In this paper, a general approach to detect 3D objects from a single image without training data was proposed. The detected objects are then integrated into a point-object SLAM framework to jointly improve the poses of the camera, the points and the object itself.

Our 3D object detection method is based on a 2D object detection as well as dimension and orientation sampling. The translation is solved by a non-linear least square system, and the selection is implemented by aligning to image features after projecting all proposals onto the image plane. Experiments on public and self collected datasets show that our detection method can be applied in various scenes and works best for large and boxy like objects. We further proposed a points-object SLAM framework with feature points and best proposals. Objects hereby provide geometry constraints and semantic information for camera pose estimation, and SLAM also provides camera pose initialization for detecting and refining 3D objects.

Future work will focus on object detection of irregular objects, such as cups, lamps, etc. While we expect only a small amount of modification for the sample process, the score process needs greater changes. A deep learning based score framework is valuable [24]. Regarding the proposed SLAM framework, the scale of the map and the object association must be solved in a robust way [25]. Finally, a sensor fusion system, combining the used RGB-image with IMU or depth information is expected to improve the results [26]. Other environment information, such as lines and planes, could also be integrated into the SLAM framework to further build a high-quality map [27].

REFERENCES

[1] Cadena, Cesar, et al. "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age." IEEE Transactions on robotics 32.6 (2016): 1309-1332.

[2] Bowman, Sean L., et al. "Probabilistic data association for semantic SLAM." 2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017.

[3] Wei, ShangGuan, et al. "Survey of connected automated vehicle perception mode: from autonomy to interaction." IET Intelligent Transport Systems 13.3 (2019): 495-505.

[4] Murthy, J. Krishna, et al. "Reconstructing vehicles from a single image: Shape priors for road scene understanding." 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017.

[5] Lee, David C., Martial Hebert, and Takeo Kanade. "Geometric reasoning for single image structure recovery." 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009.

[6] Mousavian, Arsalan, et al. "3D bounding box estimation using deep learning and geometry." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[7] Yang, Shichao, and Sebastian Scherer. "CubeSLAM: Monocular 3D object SLAM." IEEE Transactions on Robotics 35.4 (2019): 925-938.

[8] Mur-Artal, Raul, and Juan D. Tardós. "Orb-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras." IEEE Transactions on Robotics 33.5 (2017): 1255-1262.

[9] Salas-Moreno, Renato F., et al. "SLAM++: Simultaneous localisation and mapping at the level of objects." Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.

[10] Choudhary, Siddharth, et al. "SLAM with object discovery, modeling and mapping." 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014.

[11] Rubino, Cosimo, Marco Crocco, and Alessio Del Bue. "3D object localisation from multi-view image detections." IEEE transactions on pattern analysis and machine intelligence 40.6 (2017): 1281-1294.

[12] Nicholson, Lachlan, Michael Milford, and Niko Sünderhauf. "QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented SLAM." IEEE Robotics and Automation Letters 4.1 (2018)

[13] Yang, Shichao, and Sebastian Scherer. "Monocular object and plane SLAM in structured environments." IEEE Robotics and Automation Letters 4.4 (2019): 3145-3152.

[14] Hosseinzadeh, Mehdi, et al. "Structure aware SLAM using quadrics and planes." Asian Conference on Computer Vision. Springer, Cham, 2018.

[15] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).

[16] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the KITTI vision benchmark suite." 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012.

[17] Song, Shuran, Samuel P. Lichtenberg, and Jianxiong Xiao. "Sun RGB-D: A RGB-D scene understanding benchmark suite." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[18] Sturm, Jürgen, et al. "A benchmark for the evaluation of RGB-D SLAM systems." 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012.

[19] Handa, Ankur, et al. "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM." 2014 IEEE international conference on Robotics and automation (ICRA). IEEE, 2014.

[20] Naiden, Andretti, et al. "Shift R-CNN: Deep monocular 3D object detection with closed-form geometric constraints." 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019.

[21] Chen, Xiaozhi, et al. "Monocular 3d object detection for autonomous driving." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[22] Xiao, Jianxiong, Bryan Russell, and Antonio Torralba. "Localizing 3D cuboids in single-view images." (2012).

[23] Choi, Wongun, et al. "Understanding indoor scenes using 3d geometric phrases." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013.

[24] Liu, Lijie, et al. "Deep fitting degree scoring network for monocular 3d object detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[25] Li, Jimmy, et al. "View-invariant loop closure with oriented semantic landmarks." 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020.

[26] Qin, Tong, Peiliang Li, and Shaojie Shen. "VINS-MONO: A robust and versatile monocular visual-inertial state estimator." IEEE Transactions on Robotics 34.4 (2018): 1004-1020.

[27] Hosseinzadeh, Mehdi, et al. "Real-time monocular object-model aware sparse SLAM." 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.