Object-based Loop Closure with Directional Histogram Descriptor

Benchun Zhou[1], Yongqi Meng[1] and Furmans Kai[1]

***Abstract*— Loop closure can effectively eliminate the accumulated error in Simultaneous Localization and Mapping (SLAM). Appearance-based localization methods tend to fail under large viewpoint changes. In this paper, we propose a monocular SLAM system with object-based loop closure against viewpoint variation to achieve global localization. Objects are represented as cuboids and inferred from 2D object observation. On this basis, we construct a semantic topology graph from the object-oriented map and propose an efficient graph matching method with a directional histogram descriptor to detect the loop. Objects are matched if they satisfy general, graph and geometry verifications. By aligning the matched objects, the accumulated errors can be corrected, and the map can be updated. Experimental results demonstrate that the proposed method shows high accuracy and robustness under large viewpoint differences.**

Keywords: object mapping, loop closure, SLAM system

I. Introduction

In the process of localization and mapping, mobile robots will produce inevitable accumulated errors because of sensor noise [1]. To elimilate this drift errors, loop closure is proposed to recognize if the place and landmarks has been visited.

Traditional appearance-based loop detection methods rely on matching image intensities with local appearance-based features to recognize the same scene [2]. These methods can establish impressive results under similar perceptual conditions. However, the indoor environment appearance changes dramatically due to the lighting condition and object occlusion. Different camera viewing conditions in the same scene lead to different observations [3]. These factors make it challenging to detect loop with only low-level appearance features.

Semantic objects are high-level semantic landmarks that can be detected regardless of viewpoints [4]; therefore, the object-oriented map inherently supports view-invariant loop detection. Inspired by this, an object-based loop closure method is proposed to achieve global localization. An example is shown in Fig. 1. Our system takes RGB images as input to builds an object-oriented map consisting of feature points and cuboid objects. The appearance-based feature points are valuable for accurately tracking camera pose locally, and semantic objects are robust under large changes in viewing angle. Then, we represent the map as a topology graph and propose a graph matching method to detect the object loop. If an object loop is detected, we compute the similarity transform and correct the map by aligning the looped objects with non-linear optimization.

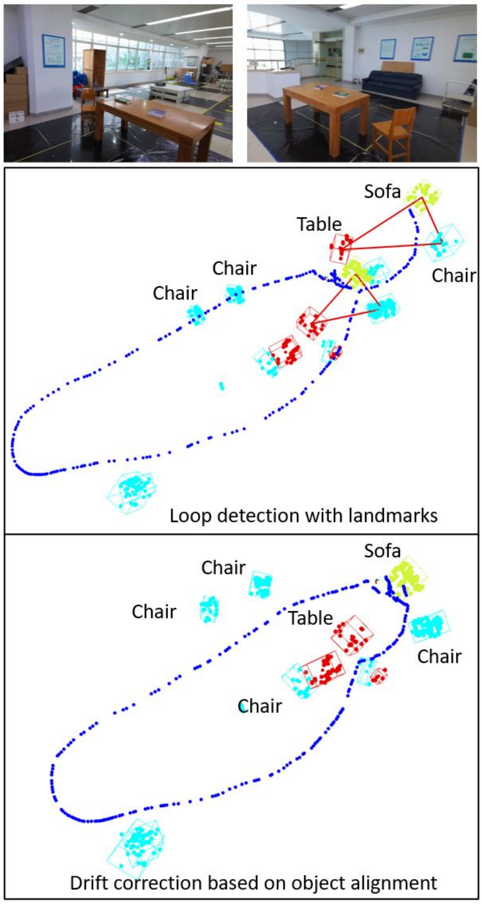

Sofa
Table
Chair
Chair
Chair
Loop detection with landmarks
Sofa
Chair
Chair
Table
Chair
Chair
Drift correction based on object alignment

Fig. 1. An example of our SLAM system with object-based loop closure. Top: Same objects are observed from two different viewpoints. Middle: An object-based loop is detected by graph matching. Bottom: The drift error is corrected by object alignment.

We evaluate our proposed object-based loop closure method on public datasets. The results show that our method can detect loop with object landmarks under different viewpoint, and the drift error can be corrected by object alignment. To summarize, the contributions of the paper are as follows:

- We present a monocular SLAM system that can build a semantic map with feature points and object landmarks.
- An object-based loop closure method based on semantic graph matching is proposed, which is robust to the change of viewpoint.
- We propose a loop correction method to align the matched objects, which can effectively correct the drift error.

*This work was supported by CSC Foundation

[1] Benchun Zhou, Yongqi Meng and Furmans Kai are with Institute for Material Handling and Logistics, Karlsruhe Institute of Technology, 76131, Germany. benchun.zhou@kit.edu

II. Related Work

A. Object Mapping

Objects are high-level entities of the environment that carry semantic and geometric information. Compared to low-level geometric features, the introduction of objects can benefit both the localization and mapping process.

Salas et al. [4] proposed a real-time object-oriented RGB-D SLAM system with prior object models, the system demonstrated high performances on object recognition and camera tracking. Sünderhauf et al. [5] took mesh-based geometric representations for objects and presented a semantic mapping system that combines object detection, instance-level segmentation, and object updates. With a similar idea, Dengler et al. [6] extended the SLAM system to be capable of online semantic mapping and object updating.

Except for specific types of object models, geometry shapes can also be used to represent general objects. Nicholson et al.[7] estimated 3D quadrics surface for objects from multiple views and used these objects to localize the camera position. Their experiments showed that quadric landmarks provide valuable information for correcting odometry errors. Yang et al. [8] represented objects as cuboids and proposed a geometric method to model cuboids from a single RGB image. The results showed that objects can provide long-range geometric and scale constraints to improve camera pose estimation and reduce monocular drift. Wu et al. [9] took cuboids as objects model and presented an outlier-robust centroid and scale estimation algorithm that can initialize accurate objects on monocular feature points.

In this paper, we focus on monocular object-level mapping and loop detection. Objects are initialized as 3D cuboids and associated with feature points, thus we can map the environment with object landmarks. On this basis, we represent the object-oriented map as a semantic graph and explore graph matching method to detect loops under large viewpoint changes.

B. Object-based Loop Closure

Loop detection methods requires a robust descriptor to represent the observation. The appearance-based methods extract local or global visual features to find the association of images, such as SIFT [10], ORB [11], GIST [12]. Global localization is then approached as an image retrieval problem. These methods show good performance when the appearance difference between images is small. In some conditions, when the viewpoint is different, the localization systems become less reliable. Instead, the object information in the scene is robust for viewpoint variation, lighting condition and occlusion. This has inspired SLAM methods to use high-level object landmarks for view-invariant loop closure.

Li et al. [13] proposed an object-based loop closure mechanism on the layout of mapped objects. This method searched for pair-wise object matches by scale, translation, and rotation consistency, thus relied heavily on object initialization results. However, scene occlusion and duplicated objects often appear in the environment, making it difficult to match the objects by exhaustive searching.

Instead, graph-based methods formulate the global localization as a graph registration problem, and convert the loop detection problem to extract the correspondences between nodes across the graphs.

Gawel et al. [14] introduced a random walks descriptor for every node that encodes the local connectivity of neighboring nodes. Based on the descriptor, the object match problem can be solved by computing the descriptor similarity between nodes. Liu et al. [15] combined random walk descriptor with graph matching and 3D alignment, the proposed object-level global localization algorithm is robust to illumination changes, view-point variation, etc. However, when the matching graphs are large, the computational complexity becomes extremely high.

Lin et al. [16] proposed an efficient graph matching method based on edit distance for robust place recognition, which demonstrated both high accuracy and robustness against drastic scene and viewpoint variation. Guo et al. [17] presented a semantic histogram-based graph matching algorithm for global localization. This algorithm achieved good performances in large viewpoint differences and can perform in real-time.

All above system are based on RGB-D images and the graph is constructed based on neighboring information. However, the scale problem in the monocular SLAM system makes it hard to construct a graph by absolute distance. By contrast, we use relative distance information to select neighboring nodes and propose a directional histogram descriptor for graph matching, which enables a novel object loop detection method for monocular SLAM systems.

III. Method

The overview of our system is shown in Fig. 2, which is built on the state-of-art ORB-SLAM 2 [18]. The system takes RGB image sequences as input to achieve camera tracking, semantic mapping and loop closure. To make the system more stable, we keep the feature-based loop closure to track the camera locally and add another object-based loop closure for view-invariant matching. To do that, we firstly initialize semantic objects with feature points, build an object-oriented map and construct a lightweight topology graph to record object information. On this basis, an object-based loop detection schedule with the directional histogram descriptor is proposed. If a loop is detected, we align looped objects, perform pose correction for camera frames and update the whole map with graph-based non-linear optimization.

A. Object Mapping

Given RGB image sequences, the semantic mapping algorithm aims to compute 3D poses of objects and produce an object-oriented map, where the object landmarks are represented as bounding cuboids (class, 3 DoF translation, 3 DoF rotation, and 3 DoF dimension). However, we just use class and translation information for object-based loop detection.

The system is built on ORB-SLAM 2 [18], where the ORB feature points are used to initialize the map and track

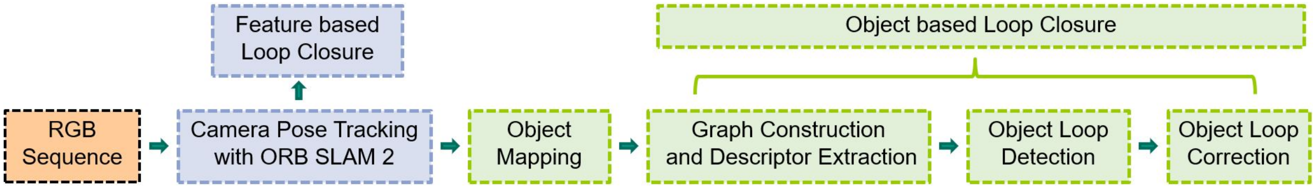

Feature based Loop Closure
Object based Loop Closure
RGB Sequence
Camera Pose Tracking with ORB SLAM 2
Object Mapping
Graph Construction and Descriptor Extraction
Object Loop Detection
Object Loop Correction

Fig. 2. Overview of our SLAM system. We add the object-based loop closure to the system, which contains four key components: object mapping (III-A), graph construction (III-B) , object-based loop detection (III-C) and loop correction (III-D).

the camera poses. For every image frame, we detect 2D bounding boxes of objects using YOLO [19] and associate them with feature points, then, we check if the detection is associated with an existing object landmark. To do that, we project object landmarks onto the current frame to calculate their 2D IoU (Intersection over Union). If it is larger than a threshold (> 0.5), the detection and associated feature points are merged into existing objects. Otherwise, a new object landmark is initialized with the detection at camera coordinate system, where the centroid is computed by the mean value of feature points X, the rotation is sampled and match with detected lines [9], and the dimension is calculated by $dim = (max(X) - min(X))/2$. In addition, an outlier-robust centroid and scale estimation algorithm from [9] is adopted to improve the estimation accuracy.

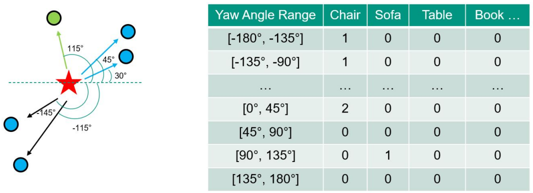
B. Directional Histogram Descriptor

We represent the object-oriented map as a topological graph $G = \{N, E\}$, where N and E denote the nodes and edges. Object landmarks are defined as nodes with corresponding semantic properties: $N_i = \{ID, Class, Centroid, Associated\ feature\ points, Descriptor, Others\}$, where ID is the serial number added to the graph in order, $Others$ contains some other information, such as rotation, dimension, color, etc. For topological edges, since it is hard to define a threshold to measure the distance between two objects in monocular SLAM, we connect each node with the closest $k (k = 6)$ neighboring nodes with $E_{ij} = \{dis_{ij}, yaw_{ij}\}$, where dis_{ij} and yaw_{ij} are relative distance and direction angle between node N_i and N_j.

We propose a directional histogram descriptor to describe each node with surrounding information. The idea comes from the Neighbor Vector descriptor [20] and the Semantic Histogram descriptor [17]. To be specific, for each node, we define the direction angle range by splitting 2π into $div (div = 8)$ parts and count all neighboring nodes regarding the yaw angle and the label. An illustration of our descriptor is shown in Fig. 3

C. Loop Detection

To detect object-based loop, we manually divide all objects into two groups according to their ID in the node, because we aim to match the most recently mapped landmarks to earlier landmarks. Let L_a and L_b be two landmark groups, we attempt to identify a sub-group of L_a that matches the nodes from L_b. To avoid false match, we check each candidate pair to meet the following verifications.



Yaw Angle Range	Chair	Sofa	Table	Book …
[-180°, -135°]	1	0	0	0
[-135°, -90°]	1	0	0	0
…	…	…	…	…
[0°, 45°]	2	0	0	0
[45°, 90°]	0	0	0	0
[90°, 135°]	0	1	0	0
[135°, 180°]	0	0	0	0

Fig. 3. An illustration of our directional histogram descriptor of the red star. The blue and green node denote chair and sofa.

General verification: We only consider the landmarks that have the same label and a large closeness. To measure the closeness of nodes a and b, we define the ID separation between them as:

$$\eta(a, b) = |ID_a - ID_b| \ (a \in L_a, b \in L_b), \tag{1}$$

we define a closeness threshold $c\_thre = 3$, if $\eta(a, b) > c\_thre$, a and b are considered as loop candidates, otherwise, they should be merged by data association.

Graph verification: To verify two candidates that have similar graph structures, we compare their directional histogram descriptor and compute the similarity score. Similar to [17], we compute the similarity score by calculating the normalized dot-product between two descriptors as follows:

$$\mathrm{Score(A, B)} = \frac{\sum_{d=1}^{n_d} A_d \times B_d}{\sqrt{\sum_{d=1}^{n_d} (A_d)^2} \times \sqrt{\sum_{d=1}^{n_d} (B_d)^2}}, \tag{2}$$

where A and B denote descriptors of nodes from two graphs. n_d is the descriptor size, which is equal to angle range parts div multiple semantic label size ($div \times n = 8 \times 6 = 48$). We get coarse matches if the similarity score is larger than the descriptor similarity threshold. ($des\_thre = 0.8$).

Geometry verification: The coarse matching candidates still contain many incorrect matches. For example, one node in L_a may match two nodes in L_b. In graph-based localization, the transformation between two graphs is rigid, and it has been shown that given at least 3 matched objects can recover the relative transform [3]. So, we establish sub-graphs as (a_1, a_2, a_3) and (b_1, b_2, b_3) from coarse matches and check if the connected edges have the scale consistency. We compute the scale ratio s of the connected edges and define a scale factor δ to measure the scale error:

$$s_{12} = \left|dis_{a_1,a_2}\right| / \left|dis_{b_1,b_2}\right|, \tag{3}$$

$$\delta = \sqrt{(s_{12}-\overline{s})^2 + (s_{23}-\overline{s})^2 + (s_{13}-\overline{s})^2}, \tag{4}$$

where $\overline{s} = 1/3 \times (s_{12}+s_{13}+s_{23})$ represents the average scale ratio. False object correspondences often lead to an unreasonable scale ratio and have a large scale error. So, only if the scale error is smaller than scale threshold($s\_thre = 0.01$), these two sub-graphs are marked as inlier matches.

Duplicated verification: We traverse all possible sub-graph combinations of coarse matches, which may cause duplications in different inlier matches. So, the last process is to check all inlier matches, add them to final matches, and remove the duplication if they exist. Having at least 3 final matches can be recognized as a loop, and the corresponding landmark information are extracted for loop correction.

D. Loop Correction

Once a candidate is accepted as a valid loop, it can be used for drift correction by object alignment. In monocular SLAM systems, we aim to compute a similarity transformation consisting (rotation R, translation t, and scaling s) where objects from group L_a can be mapped to corresponding objects from group L_b as: $O_a = sR{*}O_b{+}t$. In fact, we found that only using the object centroid may cause large errors, instead, we take the associated map points and apply a scaled ICP [21] to calculate the similarity transform. Given two looped sub-graphs $(a_1, a_2, \cdots, a_n)$ and $(b_1, b_2, \cdots, b_n)$, we first compute the scaling using all matched object centroid:

$$s = \frac{1}{n}\sum \frac{|dis_{a_1,a2}|}{|dis_{b_1,b_2}|} + \cdots + \frac{|dis_{a_n,a1}|}{|dis_{b_n,b_1}|}. \tag{5}$$

Then, we scale all map points from sub-group $(b_1, b_2, \cdots, b_n)$, and obtain inlier map point correspondences, and compute the rigid transform. The rotation R and translation t is calculated by minimizing the sum of squared error:

$$(R^*, t^*) = \operatorname*{argmin}_{R,t} \sum_{k=1}^{N_p} \left\| p_{a,k} - sR * p_{b,k} - t \right\|^2, \tag{6}$$

where $p_{a,k}$ and $p_{b,k}$ are the correspondence map points of two sub-graphs after RANSAC rejection, and N_p is the matched points number. After obtaining the similarity transform, we perform the transform to the recently mapped landmarks, merge the duplicated objects, and correct the camera poses that can observe these objects. Finally, we update the whole semantic map with non-linear least square optimization introduced in [18].

IV. EXPERIMENTS

A. Datasets

We evaluate our method on publicly available TUM RGB-D dataset [22]. This dataset focus on visual SLAM in indoor scenes, which records environment data with RGB-D cameras and provides ground truth with motion capture equipment. We select some sequences that contain object loops for experiments.

USTC RGB-D dataset [16] is another suitable dataset that has object-based loop under large viewpoint different. The indoor RGB-D sequences data are captured by the Azure Kinect DK sensors and the ground truth is estimated by VINS-Mono [23] with high accuracy.

B. Object Mapping Performance

We select several sequences to evaluate our object mapping method, the results are shown in Fig. 4. The first two lines are $fr2\_desk$ and $fr3\_long\_office$ sequence from TUM RGB-D dataset, and the last line comes from USTC dataset, namely $ustc\_01$ sequence. The first column shows images with detected feature points and 2D bounding boxes, the second column displays the corresponding point map built by ORB-SLAM 2 [18], and the object-oriented maps consisting of objects are shown in the third column. As we can see, our method can map the environment with object landmarks. Due to the sparsity of feature points, the rotation and dimension of mapped objects may contain errors, but the centroid of objects is relatively accurate, which can be used for graph construction and matching.

C. Loop Detection Performance

In our experiments, the loop detection can be considered as graph matching between two object-oriented maps. So, we decompose each sequence into several segments $Q = \{q_1, q_2, \cdots, q_n\}$ and select two segmentations $(q_i, q_j \in Q)$ that contain the same place from different viewing angles to verify our loop detection method. For each segmentation, our system builds an object-oriented map, constructs the object graph, and attempts to match it to another graph with the directional histogram descriptor. Fig. 5 presents two matched object graphs in the template images and maps. The viewing angle differences are more than $90°$. From the figure, we can observe that our loop detection method can detect object loop and is robust to the changes of large viewpoint.

D. Localization Performance

We evaluate our SLAM system with object-based loop closure in different datasets. An example from $ustc\_02$ sequence can be found in Fig. 1. A loop is detected by matching 3 objects landmarks from different viewpoints, then, we correct the drift error by aligning the looped object and update the whole semantic map. A corresponding video and part of our code are open-source. [1] A comparative analysis is performed between ORB-SLAM 2 [18] and our method. To compare localization accuracy, in monocular SLAM, the estimated camera trajectory is scaled and aligned with the ground truth. The RMSE (Root Mean Square Error) of these trajectories are presented in Table I, while the estimated trajectories of different loop closure algorithms are shown in Fig. 6.

[1] https://github.com/benchun123/object-based-loop-closure.git

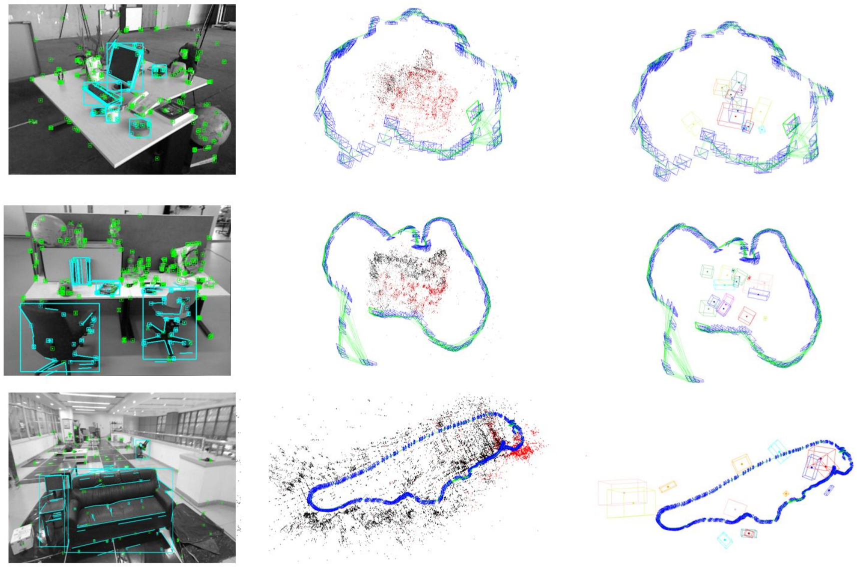
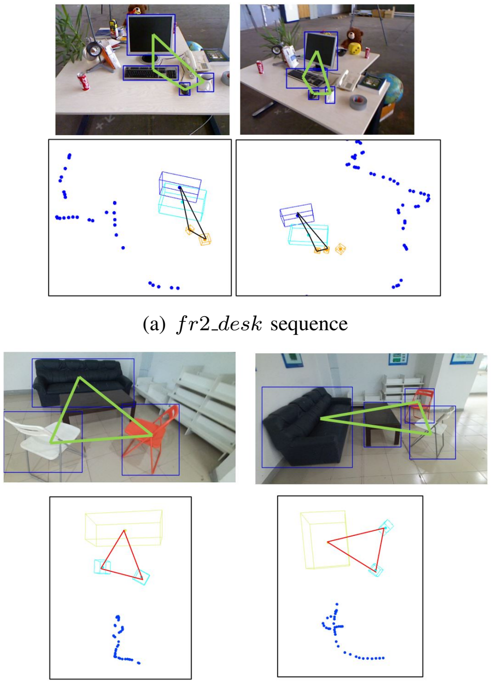
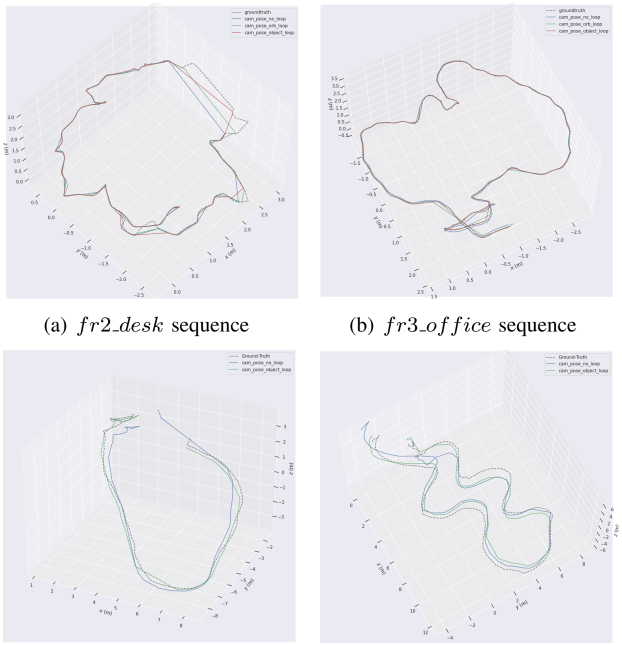

(a) image (b) feature point map (c) object-oriented map

Fig. 4. Results of semantic mapping on $fr2\_desk$, $fr3\_office$ and $ustc\_01$ sequences. (a) the images with 2D detections and associated feature points. (b) the feature point map built by ORB-SLAM 2 [18]. (c) the object-oriented map built by our system.


(a) $fr2\_desk$ sequence

(b) $ustc\_07$ sequence

Fig. 5. Two examples of object-based loop detection in different viewpoint. (a) $fr2\_desk$ sequence. (b) $ustc\_07$ sequence. Top: matched object graph on the image. Bottom: matched object graph in object-oriented map.



(a) $fr2\_desk$ sequence (b) $fr3\_office$ sequence

(c) $ustc\_02$ sequence (d) $ustc\_07$ sequence

Fig. 6. Trajectory comparison in several sequences of TUM and USTC RGB-D dataset. (a) $fr2\_desk$ (b) $fr3\_office$ (c) $ustc\_02$ (d) $ustc\_07$

The TUM datasets [22] allow the camera to detect not only object-based but also feature-based loop, because it reimage the same surface without large changes in viewpoint. Compared to loop closure method with only feature-based loop, our method does not significantly enhance localization accuracy. However, in USTC datasets [16] with huge image appearance difference in the end, ORB-SLAM 2 failed to detect the loop and resulted in large drift error. By contrast, object-based method can match the object even in 180° viewpoint difference and achieve a great improvement in localization.

E. Computation Time

All methods are implemented in C++ and evaluated on a laptop computer (i7-8565U 1.80 GHz CPU, 16 GB RAM,

TABLE I

THE TRANSLATION ERROR OF LOCALIZATION ON DIFFERENT DATASET

Datasets	ORB SLAM without loop closure (m)	ORB SLAM with feature-based loop closure (m)	Ours (m)
fr2_desk	0.012499	**0.007723**	0.008851
fr3_office	0.034488	**0.012362**	0.013587
utsc_01	1.515137	1.515137	**0.268989**
utsc_02	0.733596	0.733596	**0.191738**
utsc_03	4.399249	4.399249	**0.742086**
utsc_07	0.733087	0.733087	**0.498279**

TABLE II

AVERAGE RUNTIME OF DIFFERENT MODULE OF OUR SYSTEM

Modules	Runtime (mSec)
Camera Pose Tracking	78.864
Object Mapping	50.946
Loop Detection	3.427
Loop Correction	293.99
Total	427.227

no GPU, Ubuntu 18.04). We select $fr3\_office$ in TUM benchmark for time analysis and show the results in Table II. Since YOLO [19] requires GPU, the computation time of the 2D object detector is not listed here. There are several modules, the camera pose tracking and object mapping modules are measured every key-frame, loop detection can be reached in real-time and only cost 3ms per key-frame. when a loop is accepted, it takes 293 ms to correct the drift error, including object alignment, global bundle adjustment, and the whole map update. Since object loop correction only happen once, it does not increase the computational burden of the system, and the whole system can run in real-time.

V. CONCLUSIONS

In this paper, a monocular object-oriented SLAM system is presented, which leverages semantic mapping, camera tracking, graph matching, and drift correction. We evaluate this proposed approach on public available dataset with different viewing angle. The results demonstrated that our method can detect object-based loop against viewpoint variation and achieve high performances in localization.

Due to the limitation of the RGB camera, 3D object detection and initialization are not accurate, much information such as dimension and color cannot be used for object matching. For future work, we will develop new methods for object detection with different sensors. Besides, we also plan to explore the potential of semantic map and design field experiment in logistic environment.

REFERENCES

[1] J. Wang, P. Wang, and Z. Chen, “A novel qualitative motion model based probabilistic indoor global localization method,” *Information Sciences*, vol. 429, pp. 284–295, 2018.

[2] J. Wang, P. Wang, D. Dai, M. Xu, and Z. Chen, “Regression forest based rgb-d visual relocalization using coarse-to-fine strategy,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4431–4438, 2020.

[3] J. Li, D. Meger, and G. Dudek, “Semantic mapping for view-invariant relocalization,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7108–7115.

[4] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, “Slam++: Simultaneous localisation and mapping at the level of objects,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1352–1359.

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

[6] N. Dengler, T. Zaenker, F. Verdoja, and M. Bennewitz, “Online object-oriented semantic mapping and map updating,” in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2020, pp. 1–7.

[7] L. Nicholson, M. Milford, and N. Sünderhauf, “Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam,” *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, 2018.

[8] S. Yang and S. Scherer, “Cubeslam: Monocular 3-d object slam,” *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[9] Y. Wu, Y. Zhang, D. Zhu, Y. Feng, S. Coleman, and D. Kerr, “Eao-slam: Monocular semi-dense object slam based on ensemble data association,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4966–4973.

[10] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[12] A. Oliva and A. Torralba, “Building the gist of a scene: The role of global image features in recognition,” *Progress in brain research*, vol. 155, pp. 23–36, 2006.

[13] J. Li, K. Koreitem, D. Meger, and G. Dudek, “View-invariant loop closure with oriented semantic landmarks,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7943–7949.

[14] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena, “X-view: Graph-based semantic multi-view localization,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1687–1694, 2018.

[15] Y. Liu, Y. Petillot, D. Lane, and S. Wang, “Global localization with object-level semantics and topology,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4909–4915.

[16] S. Lin, J. Wang, M. Xu, H. Zhao, and Z. Chen, “Topology aware object-level semantic mapping towards more robust loop closure,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7041–7048, 2021.

[17] X. Guo, J. Hu, J. Chen, F. Deng, and T. L. Lam, “Semantic histogram based graph matching for real-time multi-robot global localization in large scale environment,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8349–8356, 2021.

[18] R. Mur-Artal and J. D. Tardós, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[20] E. Stumm, C. Mei, S. Lacroix, J. Nieto, M. Hutter, and R. Siegwart, “Robust visual place recognition with graph kernels,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4535–4544.

[21] T. Zinßer, J. Schmidt, and H. Niemann, “Point set registration with integrated scale estimation,” in *International conference on pattern recognition and image processing*, 2005, pp. 116–119.

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

[23] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.