2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Resolving Loop Closure Confusion in Repetitive Environments for Visual SLAM through AI Foundation Models Assistance

Hongzhou Li, Sijie Yu, Shengkai Zhang, Guang Tan

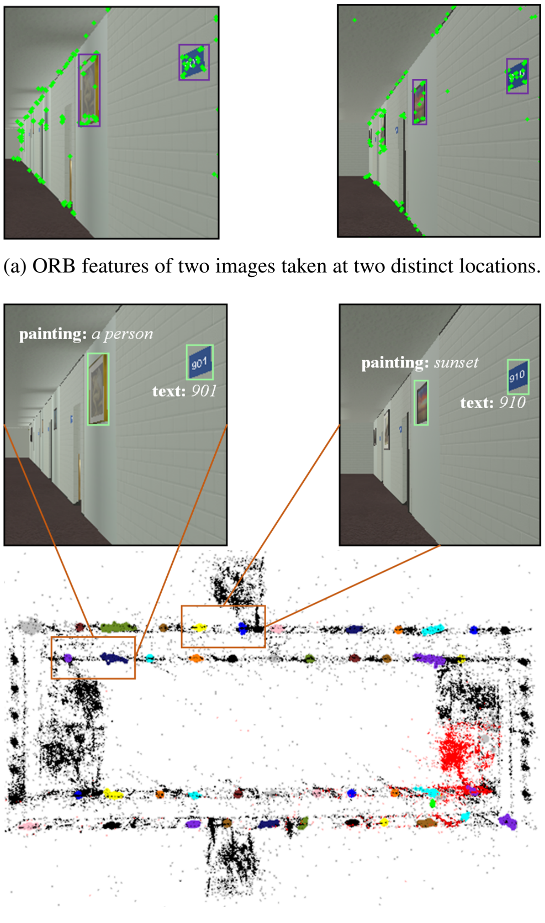
***Abstract*— In visual SLAM (VSLAM) systems, loop closure plays a crucial role in reducing accumulated errors. However, VSLAM systems relying on low-level visual features often suffer from the problem of perceptual confusion in repetitive environments, where scenes in different locations are incorrectly identified as the same. Existing work has attempted to introduce object-level features or artificial landmarks. The former approach struggles to distinguish visually similar but different objects, while the latter is both time-consuming and labor-intensive. This paper introduces a novel loop closure detection method that leverages pretrained AI foundation models to extract rich semantic information about specific types of objects (e.g., door numbers), referred to as *semantic anchors*, that help to distinguish similar scenes better. In settings such as office buildings, hotels, and warehouses, this approach helps to improve the robustness of loop closure detection. We validate the effectiveness of our method through experiments conducted in both simulated and real-world environments.**

I. INTRODUCTION

The Visual Simultaneous Localization and Mapping (VSLAM) system uses the camera to explore the environment and simultaneously build a model of the surroundings. Early work by Davison et al. [1] and Klein et al. [2] laid the foundation for VSLAM, followed by extensive subsequent efforts that achieved more accurate and stable results. These approaches depend on low-level visual features and struggle in environments with a high degree of repetitive elements.

To address this issue, researchers have proposed several solutions: (1) using object-level features [3][4][5]; (2) introducing artificial landmarks into the environment [6] [7] [8] [9][10]; (3) incorporating non-visual environmental information, such as magnetic fields [11] [12], or radio-frequency signals [13]. The first approach encounters challenges in scenes where objects have similar appearances and poses, while the last two require pre-installation or assume extra sensors, leading to additional costs.

In this paper, we propose a novel loop-closure detection method for common repetitive environments like office buildings, libraries, and warehouses. Our method leverages pretrained AI foundation models to obtain natural language descriptions of specific objects in the environment, such as door numbers, shelf numbers, directional signs, etc. Traditional descriptors based on pixel features often fall short in distinguishing between various instances of these objects (e.g., door numbers ”901” and ”910”). However, these objects carry vital semantic information that identifies distinct areas. Furthermore, they serve as stable and consistent elements in the surroundings, ensuring reliability in mapping and localization. We term these semantically distinctive and stable objects as *semantic anchors* or simply *anchors*, see Figure 1 for an example. In our approach, we make use of two foundation models, namely Blip-2 [14] and



(a) ORB features of two images taken at two distinct locations.

(b) Semantic anchors at various locations.

Fig. 1: Distinguishing between two similar scenes in a repetitive environment that comprises a corridor and multiple doors. (a) Using conventional ORB features, it is difficult to distinguish between the two scenes. (b) Semantic objects, each annotated with a textual description, enable clear distinction between the scenes. The colored point clouds on the map represent various semantic anchors.

Hongzhou Li and Sijie Yu are with School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, China (e-mails:{lihzh55, yusj8}@mail2.sysu.edu.cn).

Shengkai Zhang is with School of Information Engineering, Wuhan University of Technology, Wuhan, China (e-mail:shengkai@whut.edu.cn).

Guang Tan (corresponding author) is with School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, China (e-mail: tanguang@mail.sysu.edu.cn.

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

ChatGPT [15], to perform semantic analysis. This eliminates the need for us to design and train specific models for individual tasks and scenarios.

To achieve loop-closure detection with semantic anchors, we need to address two problems. First, it is non-trivial to determine the semantic anchors and extract their rich semantic information to achieve object-level data association. Second, as the system operates, the semantic anchors are subject to accumulated errors, potentially causing mismatch of anchors. Thus we require precise position estimation.

For the first problem, we use the AI foundation models to generate natural language descriptions for both text and images. These models not only create scene descriptions but also can assess their quality and similarity. To address the second problem, we devise a local anchor map constructed based on co-visibility relationships. When comparing two local anchor maps, we take advantage of the relative positions between anchors, as these are less susceptible to cumulative errors.

We tested our method in both simulated and real-world environments and compared it with the open-source ORB-SLAM3 [16] system in terms of loop-closure detection and trajectory error. Experimental results indicate that our method performs successfully in loop-closure detection when semantic anchors are observed. Due to correct loop closures, our method also achieves higher localization accuracy.

The main contributions of this paper are as follows:

- Introducing the concept of semantic anchors and utilizing foundation models to obtain their natural language descriptions. Based on this, we implemented an object-level data association method.
- Proposing a new semantic loop-closure detection method that constructs a local map using co-visibility relationships to distinguish similar scenes by comparing the textual descriptions of objects and their relative locations in the local map.

II. RELATED WORK

Semantic SLAM incorporates semantic information about the objects in the environment. The main advantage of Semantic SLAM is that it provides a more comprehensive understanding of the scene and its elements. SLAM++ [17] pioneered object-level SLAM, in which camera and object poses are jointly optimized, assuming access to prior object models. Cubeslam [18], Quadricslam [19], Eao-slam [20], and So-slam [21] further eliminated the need for prior object models, and approximated objects using standardized models such as cubes and ellipsoids. Notably, these approaches did not address loop-closure detection.

Hu et al. [3], Li et al. [4], and Qian et al. [5] integrated low-level features with semantic object pose information for loop-closure detection, relying on the relative positions of objects to differentiate similar scenes. However, they did not take into account the specific characteristics of individual objects. In highly repetitive settings such as hotel corridors or warehouses, considering only relative object positions is inadequate for scene discrimination.

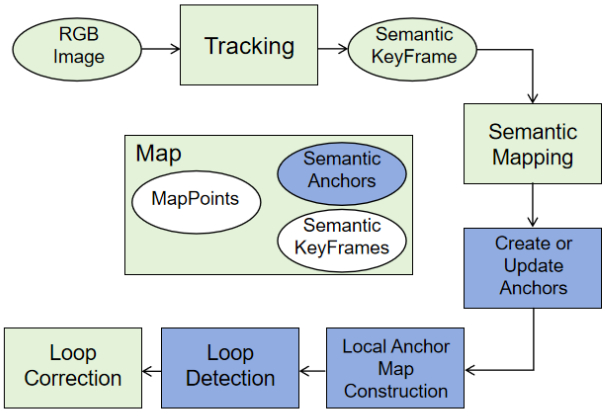


Fig. 2: An overview of proposed system.

Textslam [22][23] primarily focused on textual objects, treating them as plane-related features. However, this approach only distinguished textual objects based on pixel-level features, disregarding their semantic content. In many environments, semantic distinctions within text may occur at the level of individual letters or strokes. Yet, such nuances can be hard to discern at the pixel level.

Another category of solutions involves using artificial landmarks, including QR codes [6] [7] [8] [9], or RFID [10]. However, these methods introduce extra infrastructure costs and manual setup. Non-visual information has been considered in several previous works, such as magnetic fields [11] [12], and radio-frequency signals [13]. Again, integrating the extra sensors entails additional expenses.

Lp-slam [24] employed OCR methods to recognize textual content in the environment. It utilized foundation models to correct and understand text meanings, and achieved object localization through SLAM's tracking module. It implemented a navigation system based on natural language interaction, introducing detailed semantic information into the SLAM system. This work primarily dealt with textual objects and focused on navigation tasks.

III. SYSTEM OVERVIEW

We have embedded our semantic loop-closure detection method into ORB-SLAM3 [16], resulting in an extended system with semantic recognition, tracking, mapping, and loop-closure capabilities. The system framework is illustrated in Figure 2, with our contributed modules highlighted in blue.

First, the system extracts feature points from the input RGB images for tracking and selects representative frames as keyframes. Next, we employ Yolov6 [25] and East [26] for object/text box detection on these keyframes, resulting in a set of detection bounding boxes. Each box is associated with the following information: location and side lengths of the box, object class, detection confidence, and map points covered by the box. We record these detection bounding boxes and their associated information in the keyframes, constructing semantic keyframes.

Based on prior knowledge of the environment to be navigated, we predefine several categories of semantic anchors, such as door numbers and directional signs. If a detected object falls in these categories, we perform object-level data association. Firstly, we associate a detection bounding box with nearby semantic anchors based on the locations of the covered map points. Upon a successful association, we update the corresponding semantic anchor. Otherwise, we construct a new semantic anchor, whose data structure contains a textual description of the object, generated using Blip-2 and ChatGPT. Subsequently, using ChatGPT, we further perform semantic matching between the newly constructed semantic anchor and other semantic anchors in the map, determining whether the current semantic anchor has been observed before. If this is the case, it triggers loop-closure detection; see Chapter IV for details.

During loop-closure detection, we first employ the bag-of-words (BoW) method [27] to select similar candidate keyframes. Then, based on co-visibility relationships, we separately construct local anchor point maps for the current keyframe and the candidate keyframes. Comparing the relative positions of semantic anchors in these two local maps allows us to further distinguish ambiguous repetitive scenes, identifying potential loop closures; see Chapter V.

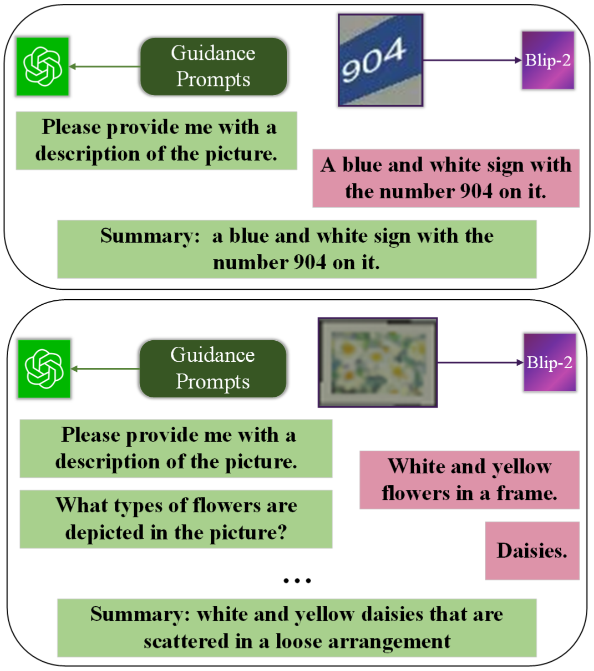
IV. SEMANTIC ANCHORS AND DATA ASSOCIATION

A. Semantic Operations on Image and Text

1) Generating image content description: We use the Blip-2 visual question and answer (QA) model [14] to generate a textual description for an image in response to a question. Typically, a simple question such as “Please describe the picture” yields only a general response describing the major components in the image. To obtain more details, we adopt the method from Zhu et al. [28], which introduces a strategy to elicit a multi-round dialogue between ChatGPT and Blip-2, as depicted in Figure 3. Throughout these interactions, ChatGPT automatically poses one question at a time regarding various aspects of an image, and Blip-2 replies to each question. Ultimately, ChatGPT generates a comprehensive summary of the image’s content. In our implementation, we input the bounding box of an object to the QA model and retrieve the corresponding textual summary.

2) Text operations: We implemented three semantic functions to operate with textual descriptions:

- `int InformationLevel(string text)`: returns an integer value between $[1,5]$ that measures the amount of distinct information contained within a given text string;
- `int SimilarityLevel(string text1, string text2)`: returns an integer value between $[1,5]$ indicating the degree of semantic similarity between two textual descriptions;
- `void MergeText(string text1, string text2)`: merge two textual descriptions into a single, semantically coherent description.



Please provide me with a description of the picture.

A blue and white sign with the number 904 on it.

Summary: a blue and white sign with the number 904 on it.

Please provide me with a description of the picture.

White and yellow flowers in a frame.

What types of flowers are depicted in the picture?

Daisies.

…

Summary: white and yellow daisies that are scattered in a loose arrangement

Fig. 3: An example of the question and answer model from [28], in which ChatGPT interacts with Blip-2 to gather information about an object within an image. The user provides some special prompts to direct ChatGPT in asking a series of questions about an object, while Blip-2 supplies a response to each question. Finally ChatGPT summarizes the answers into a cohesive statement.

TABLE I: Evaluating semantic similarity between two textual descriptions using ChatGPT.

Agent	I’m going to give you a pair of phrasal descriptions of objects that have different degrees of similarity. You need to rank the similarity on a five-point scale, from lowest to highest. For example, {a gray pillow} has similarity degree 1 to {a green tree}; {a red ball} has similarity degree 2 to {a black and white ball}; {a red ball} has similarity degree 3 to {a basketball}; {a black and white ball} has a degree of similarity 4 to {a football}; {A painting depicting a mysterious smile of a European woman} has a similarity of 5 to {Mona Lisa}. Answer the score only.
	Sure, I can help you with that. Please provide me with the pair of phrasal descriptions you would like me to rank.
...	...
Agent	Let’s try {a black cat}and{a white cat}, and give me the score only.
	2
Agent	Let’s try {some purple balls }and {grapes}.
	5

Here we provide an example illustrating the implementation of the `SimilarityLevel` function using ChatGPT API. As shown in Table I, we initially provide several manually designed examples to guide ChatGPT in learning how to assess the similarity of two object descriptions, with similarity levels ranging from 1 to 5, where higher values indicate greater similarity. Then, we provide the two textual descriptions to ChatGPT and ask for similarity assessment. If the similarity score is 4 or higher, we conclude that the two textual descriptions describe the same object.

3) Asynchronous operation: Operations with the foundation models are conducted asynchronously, with a dedicated thread responsible for acquiring and updating textual descriptions. Since semantic anchors are only applied in loop-closure detection, which does not demand high real-time performance, the impact of additional latency incurred is small.

B. Semantic Anchor Construction

Semantic anchors can be considered as map objects with textual descriptions. Each semantic anchor includes the following information:

- Class of detected object, such as door numbers or paintings.
- Set of keyframes where this object has been observed.
- Set of map points observed on this object.
- Bounding box for the object.
- Textual description of content.

The textual description is obtain with the QA model as described earlier. Normally the generated description is accurate, however if the image is low-quality, or captured from a poor angle, the QA model may generate vague or uninformative descriptions. We call the `InformationLevel` function to access its quality. If the returned value is no smaller than 4, we treat the object as a valid anchor.

Each semantic anchor always maintains an image crop defined by the object bounding box. We only request textual descriptions for the crop when its area is sufficiently large or it has not been updated for a certain period of time.

C. Object-Level Data Association

Object-level data association is divided into two cases. The first case involves associating detection bounding boxes from keyframes and the map points contained within them with existing semantic anchors in the map. The second case pertains to associating the current anchor with existing anchors in the map.

1) Object tracking: Assuming that an object i is detected within the current keyframe k. The object's detection bounding box z_i is associated with the following information: object class and detection confidence, current keyframe index, geometric information of the bounding box, and the set of map points it covers, denoted as P_i. Define the set of co-visibility frames for the current keyframe k as K_k^{cov}, which includes keyframes that share a certain proportion of map points with keyframe k. Due to the co-visibility relationship, keyframes in K_k^{cov} have relatively small accumulative pose errors and therefore provide accurate localization information.

We attempt to match z_i with semantic anchors in the map that belong to the same class as i and have been observed by keyframes in K_k^{cov}. If at least a fraction σ of the map points in z_i are covered by the bounding box of a semantic anchor, then z_i is considered a match of that semantic anchor. $\sigma \in (0,1)$ is a threshold related to tracking sensitivity, and is set to 0.2 in our case. Upon successful matching, we incorporate the keyframe k and its set of map points P_i into the data structure associated with the matched semantic anchor.

2) Semantic anchor matching: If a detection bounding box z_i cannot be matched with an existing semantic anchor, we create a new semantic anchor based on z_i and then attempt to match it with existing anchors in the map. The matching of anchors relies on the semantic similarity of their textural descriptions.

If two images contain the same object from similar perspectives, the QA often generates the same textual description. Therefore, if two anchors have identical textual descriptions, we consider them matched anchors. In cases where the QA model produces somehow different descriptions, we call the `SimilarityLevel` function to obtain a similarity score between two descriptions. If the score is equal to or greater than 4, we consider the two anchors to be matched.

For two matched semantic anchors, if their keyframes share a co-visibility relationship and have close bounding boxes, the two semantic anchors are merged. This involves combining their sets of keyframes and map points and then recalculating the bounding box based on the new set of map points. For the textual descriptions, we invoke the `MergeText` function to generate a cohesive textual description.

If there is no co-visibility relationship between the two matched semantic anchors, it indicates that the agent has revisited the same semantic anchor, suggesting a potential loop closure. We set a matching flag for these two semantic anchors, and they are treated as the same during loop closure detection. If a loop closure is confirmed and executed, the two matched semantic anchors are then merged.

V. Loop Closure Detection

To determine whether the current keyframe k forms a loop closure with another keyframe in the map, the system performs three checks sequentially. First, we exclude keyframes in the map that share a co-visibility relationship with k. Next, we calculate the bag-of-words (BoW) vector similarity scores between k and the other keyframes in the map using BoW algorithms. Then, for the keyframes with the highest similarity scores, we apply the semantic anchor-based loop closure detection method, which will be described in more detail in this section, to eliminate keyframes that do not match k semantically.

For the keyframes that pass all the previous filtering criteria, we calculate the similarity transformations between k and these keyframes and count the number of matching feature points after the projection transformation. Keyframes with the highest numbers of matching feature points are confirmed as forming a loop closure with the current keyframe.

A. Local Anchor Map Construction

To mitigate the impact of cumulative errors on the localization of semantic anchors, we have designed local anchor maps based on direct or indirect co-visibility relationships. A local anchor map is centered around a keyframe and maintains the information of all semantic anchors near that keyframe. By comparing the local anchor maps of two

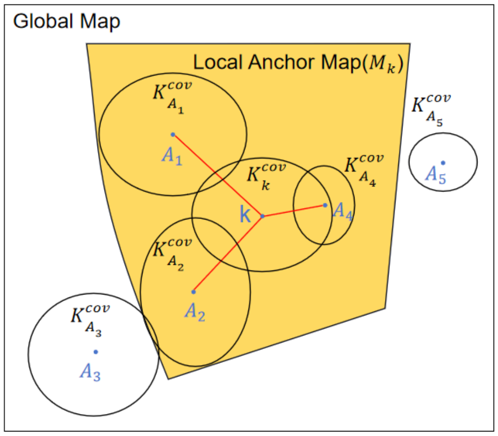


Fig. 4: A local anchor map (yellow region). Ellipses represent the sets of co-visibility frames for anchors, each with a center keyframe (blue points). The relative positions represented by red lines are used in distinguishing two local anchor maps.

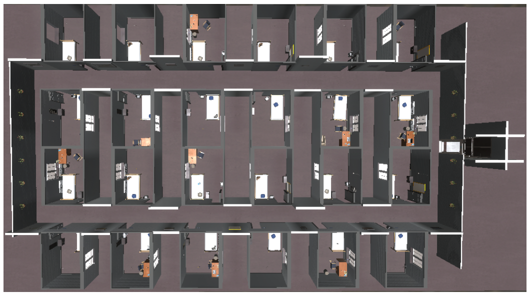


Fig. 5: Top view of the simulated environment.

TABLE II: Loop closure detection accuracy.

Dataset	Metrics	Ours	ORB-SLAM3
SE1	Precision	100%	16.7%
	Recall	100%	100%
SE2	Precision	100%	19.4%
	Recall	100%	100%

keyframes, we can confirm whether the two keyframes are in the same location.

Given a keyframe k, we now construct its *local anchor map* M_k as follows. For anchor A_i, let K_{A_i} be the set of keyframes that observe it. Recall that the set of co-visibility frames for some keyframe m is denoted K_m^{cov}, so the union of all sets of co-visibility frames for the keyframes in K_{A_i} can be defined as $K_{A_i}^{cov} = \bigcup_{m \in K_{A_i}} K_m^{cov}$, which represent the local context of anchor A_i in terms of vision overlap. If $K_{A_j}^{cov}$ and K_k^{cov}, the co-visibility frames for the current keyframe k, have a non-empty intersection, then we add A_i into the keyframe k's local anchor map M_k. Figure 4 depicts the structure of M_k, centered around keyframe k and having a number of nearby anchors incorporated.

B. Comparison of Anchor Maps

First, we calculate the angle and distance of each anchor with respect to the central keyframe in the local anchor map. Using the data association method proposed earlier for semantic anchors, it is easy to confirm which semantic anchors appear in both local anchor maps. Assume the sets of anchors in two anchor maps, M_a and M_b, are denoted as $\mathbf{A}_a$ and $\mathbf{A}_b$ respectively. For two anchors $A_i \in \mathbf{A}_a$ and $A_j \in \mathbf{A}_b$, let $x(i,j)$ represent the matching score between A_i and A_j:

$$x(i,j) = \begin{cases} 1, & \text{if } A_i \text{ matches } A_j, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Let θ_m and d_m represent the angle and distance of anchor A_m with respect to its own central keyframe in the local anchor map. If anchors A_i and A_j have similar textual descriptions and are located close to each other in their respective local anchor maps, while satisfying $|\theta_i - \theta_j| < \pi/3$ and $|d_i - d_j| < 1$, we consider A_i and A_j to be a match.

Closer semantic anchors are more valuable as references for localization than those farther away. We set the weight of semantic anchor A_m in the local anchor map as $1/d_m$. The similarity S between two local anchor maps can be expressed as the summed weight of matching anchor pairs between the two maps divided by the aggregate weight of all anchors in both maps:

$$S = \frac{\sum_{i=1}^{|\mathbf{A}_a|} \sum_{j=1}^{|\mathbf{A}_b|} [x(i,j) \cdot (\frac{1}{d_i} + \frac{1}{d_j})]}{\sum_{i=1}^{|\mathbf{A}_a|} \frac{1}{d_i} + \sum_{j=1}^{|\mathbf{A}_b|} \frac{1}{d_j}} \tag{2}$$

If S is greater than a certain threshold, we consider the two local anchor maps to be a match. We set this threshold to 0.3. The central keyframes of the two maps pass the semantic loop detection. In the next step, further calculations for a similarity transformation are performed to ultimately confirm whether they form a loop closure.

VI. EXPERIMENTS

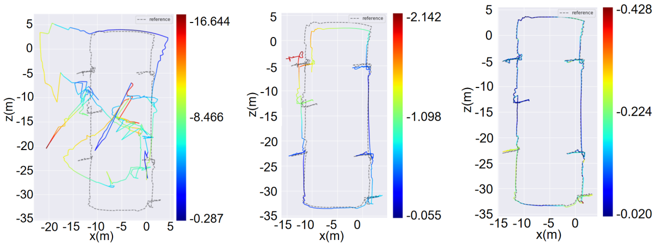
Due to the lack of highly repetitive scenes in publicly available datasets, we created a simulation environment using Unity, mimicking a hotel scenario. The ground truth for the robot's motion trajectory can be directly obtained from Unity's internal data, while the SLAM system only receives 640×480 RGB images collected by the simulated robot as its input. To validate the effectiveness of our method in the real world, we captured a sequence of images from a certain floor in a classroom building using a handheld RealSense D435i camera. We conducted tests in both the simulated and real-world environments and compared our proposed method with ORB-SLAM3 [16] in terms of loop closure correctness and localization accuracy.

A. Results from Unity simulation environment

As shown in Figure 5, the simulated hotel environment consists of 24 similarly configured rooms, each having a door number adjacent to the door. Decorative paintings are present on the walls of the long corridor. The circular corridor is approximately 100 meters long and 2.2 meters wide. The robot starts its journey from one of the rooms, traverses the

TABLE III: Location accuracy in terms of absolute pose error (m)

Dataset	Sequence	Ours	ORB-SLAM3	ORB-VO
SE1	00	0.081	11.499	0.220
	01	0.189	13.498	0.633
	02	0.083	13.539	0.223
	03	0.163	12.645	0.371
	04	0.179	12.452	0.340
	05	0.109	12.770	0.671
SE2	00	0.146	7.205	0.870
	01	0.160	9.851	0.662
	02	0.240	10.298	0.633
	03	0.448	12.792	0.961
	04	0.112	7.825	0.501
	05	0.141	9.676	0.361



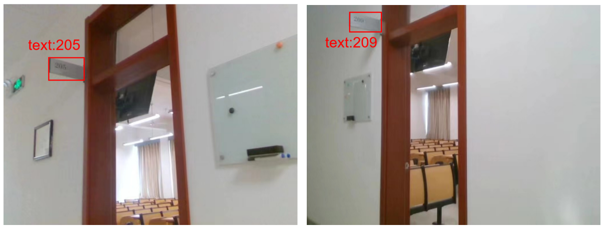
(a) (b) (c)

Fig. 6: (a) Trajectory estimated by ORB-SLAM3 compared with ground truth (gray dashed line). (b) Trajectory estimated by ORB-SLAM3 with loop closure disabled. (c) Trajectory estimated with our method.

corridor, explores several rooms en route, and finally returns to the initial room. We conducted test to evaluate the SLAM system's loop detection capability, particularly when the robot visited the rooms, to see if it would mistakenly identify one room as another. We collected two datasets, SE1 and SE2, corresponding to two environments with different room layouts. Each dataset comprises six sequences, representing journeys through six rooms arranged in different orders. The results in Table II show that our method outperforms ORB-SLAM3 in terms of both accuracy and recall.

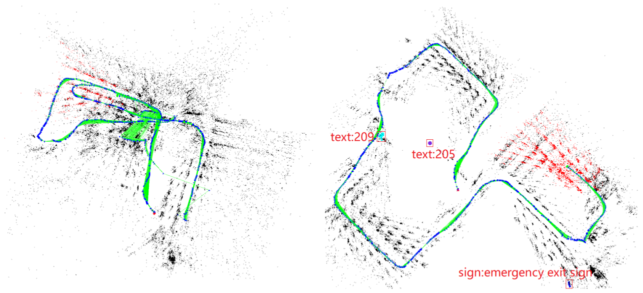
Using the pose values from Unity as ground truth, we calculated the absolute pose error of the trajectory. Since ORB-SLAM3 consistently identified incorrect loops in the tests, resulting in distorted trajectories, calculating absolute pose errors in this case would not be meaningful. Therefore, we calculated the absolute pose error of ORB-SLAM3's trajectory estimate with loop closure disabled. The results in Table III show that our method achieves better localization accuracy than ORB-SLAM3 due to more accurate loop detection. The entry "ORB-VO" refers to the result of ORB-SLAM3 with loop closure disabled.

Figure 6 visualizes the trajectory results of dataset SE2, sequence 00 from Table 1 under different methods. Due to multiple false loop detections, ORB-SLAM3 exhibits significant differences between the estimated trajectory and the ground truth. With loop closure disabled, the trajectory estimation of ORB-SLAM3 improves to some extent but is still affected by accumulated errors. Our method successfully distinguishes all similar regions, identifies true loop closures, and effectively reduces accumulated errors through loop closure, resulting in the most accurate trajectory estimates.



(a) (b)

Fig. 7: Images captured in a real-world classroom building. The two different classrooms look similar when viewed from outside, leading to incorrect loop closures reported by ORB-SLAM3. In contrast, our proposed method utilizes doorplates as semantic anchors to accurately distinguish between the two classrooms.



(a) (b)

Fig. 8: (a) An incorrect map of classrooms generated by ORB-SLAM3, where the classrooms are mixed up due to false loop detection. (b) The map produced by our method, which clearly distinguishes the classrooms with the help of semantic descriptions.

B. Results from real world

As shown in Figure 7, the real-world dataset was captured by a person walking through multiple classrooms using a handheld RealSense D435i camera. There were no modifications to the classroom layouts during the experiment. Figure 8 shows that our method successfully utilized classroom doorplates and exit signs to create semantic anchors, accurately distinguishing different classrooms and constructing clear and accurate trajectories and maps. In contrast, ORB-SLAM3 incorrectly identified three different classrooms as the same one, resulting in erroneous loop closures that caused significant distortion in the estimated trajectory and the intertwining of the three classrooms in the map.

VII. CONCLUSIONS

This paper introduces the use of semantically distinguishable objects, called semantic anchors, in repetitive environments to distinguish visually similar but distinct areas, enabling successful loop closure detection. Experimental results show that in these environments, ORB-SLAM3 often performs incorrect loop closures, while our method identifies loops correctly and achieves higher localization accuracy.

ACKNOWLEDGEMENTS

This work is supported in part by Shenzhen Baisc Research Fund under grant JCYJ20200109142217397.

References

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[2] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pp. 225–234, IEEE, 2007.

[3] M. Hu, S. Li, J. Wu, J. Guo, H. Li, and X. Kang, "Loop closure detection for visual slam fusing semantic information," in *2019 Chinese Control Conference (CCC)*, pp. 4136–4141, 2019.

[4] 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)*, pp. 7943–7949, 2020.

[5] Z. Qian, J. Fu, and J. Xiao, "Towards accurate loop closure detection in semantic slam with 3d semantic covisibility graphs," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2455–2462, 2022.

[6] B. Pfrommer and K. Daniilidis, "Tagslam: Robust slam with fiducial markers," *arXiv preprint arXiv:1910.00679*, 2019.

[7] R. Munoz-Salinas and R. Medina-Carnicer, "Ucoslam: Simultaneous localization and mapping by fusion of keypoints and squared planar markers," *Pattern Recognition*, vol. 101, p. 107193, 2020.

[8] L. E. Ortiz-Fernandez, E. V. Cabrera-Avila, B. M. d. Silva, and L. M. Gonçalves, "Smart artificial markers for accurate visual mapping and localization," *Sensors*, vol. 21, no. 2, p. 625, 2021.

[9] S. Xu, Y. Dong, H. Wang, S. Wang, Y. Zhang, and B. He, "Bifocal-binocular visual slam system for repetitive large-scale environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–15, 2022.

[10] A. Kleiner, J. Prediger, and B. Nebel, "Rfid technology-based exploration and slam for search and rescue," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4054–4059, 2006.

[11] Z. Wu, Y. Yue, M. Wen, J. Zhang, J. Yi, and D. Wang, "Infrastructure-free hierarchical mobile robot global localization in repetitive environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[12] Z. Wu, W. Wang, J. Zhang, Q. Lyu, H. Zhang, and D. Wang, "Global localization in repetitive and ambiguous environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12374–12380, 2023.

[13] S. Zhang, S. Tang, W. Wang, T. Jiang, and Q. Zhang, "Conquering textureless with rf-referenced monocular vision for mav state estimation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 146–152, 2021.

[14] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[15] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730–27744, 2022.

[16] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[17] 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*, pp. 1352–1359, 2013.

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

[19] 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.

[20] 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)*, pp. 4966–4973, IEEE, 2020.

[21] Z. Liao, Y. Hu, J. Zhang, X. Qi, X. Zhang, and W. Wang, "So-slam: Semantic object slam with scale proportional and symmetrical texture constraints," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4008–4015, 2022.

[22] B. Li, D. Zou, D. Sartori, L. Pei, and W. Yu, "Textslam: Visual slam with planar text features," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2102–2108, IEEE, 2020.

[23] B. Li, D. Zou, Y. Huang, X. Niu, L. Pei, and W. Yu, "Textslam: Visual slam with semantic planar text features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[24] W. Zhang, Y. Guo, L. Niu, P. Li, C. Zhang, Z. Wan, J. Yan, F. U. D. Farrukh, and D. Zhang, "Lp-slam: Language-perceptive rgb-d slam system based on large language model," *arXiv preprint arXiv:2303.10089*, 2023.

[25] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[26] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 5551–5560, 2017.

[27] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[28] D. Zhu, J. Chen, K. Haydarov, X. Shen, W. Zhang, and M. Elhoseiny, "Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions," *arXiv preprint arXiv:2303.06594*, 2023.