arXiv:submit/5191801 [cs.RO] 24 Oct 2023

Active Robot Vision for Distant Object Change Detection: A Lightweight Training Simulator Inspired by Multi-Armed Bandits

Terashima Kouki[1], Tanaka Kanji[1], Yamamoto Ryogo[1], and Tay Yu Liang Jonathan[1]

***Abstract*— In ground-view object change detection, the recently emerging mapless navigation has great potential to navigate a robot to objects distantly detected (e.g., books, cups, clothes) and acquire high-resolution object images, to identify their change states (no-change/appear/disappear). However, naively performing full journeys for every distant object requires huge sense/plan/action costs, proportional to the number of objects and the robot-to-object distance. To address this issue, we explore a new map-based active vision problem in this work: "Which journey should the robot select next?" However, the feasibility of the active vision framework remains unclear; Since distant objects are only uncertainly recognized, it is unclear whether they can provide sufficient cues for action planning. This work presents an efficient simulator for feasibility testing, to accelerate the early-stage R&D cycles (e.g., prototyping, training, testing, and evaluation). The proposed simulator is designed to identify the degree of difficulty that a robot vision system (sensors/recognizers/planners/actuators) would face when applied to a given environment (workspace/objects). Notably, it requires only one real-world journey experience per distant object to function, making it suitable for an efficient R&D cycle. Another contribution of this work is to present a new lightweight planner inspired by the traditional multi-armed bandit problem. Specifically, we build a lightweight map-based planner on top of the mapless planner, which constitutes a hierarchical action planner. We verified the effectiveness of the proposed framework using a semantically non-trivial scenario "sofa as bookshelf".**

I. Introduction

Ground-view change detection plays an important role for a robot to detect inconsistencies between the environment model ("map") and live scenes, in emerging application domains such as lifelong learning [1] and long-term autonomy [2]. Unlike typical vision setups such as parallel projection satellite imagery [3], the problem is complicated by nonlinear perspective projection and visual uncertainties (e.g., depth ambiguity, aspect differences, occlusion) [4].

One of the open issues in ground-view change detection is distantly detected object changes. Since distant objects (e.g., books, cups, clothes) are typically observed as spatially low-resolution object images, they often look significantly different than they were seen during training and are misrecognized as change even with state-of-the-art object detection models. An example of such misrecognition is shown in Fig. 1, where a book is misrecognized as a box.

In our observation, the recently emerging point-goal navigation [5] and other mapless navigation techniques [6]–[9] have great potential to navigate a robot to distant objects and acquire high-resolution object images, to identify their change states (no-change/appear/disappear). Specifically, these recent navigation techniques act as a reasonably efficient and lightweight action planner for purely vision-based long-distance navigation, which has traditionally been considered a computationally difficult problem.

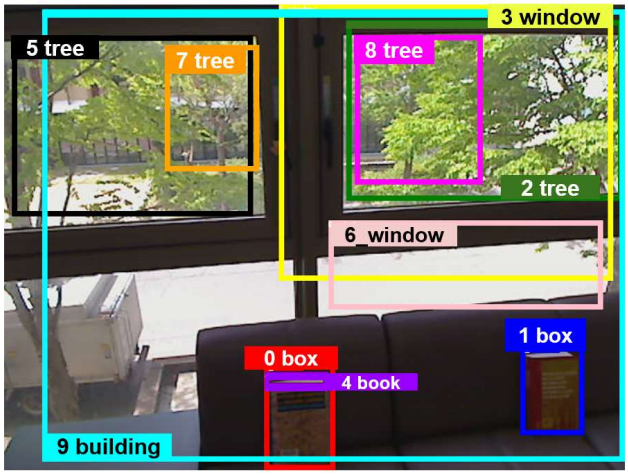

Fig. 1. Distant objects are often misrecognized as unknown objects (i.e., appearance change) due to their low resolution.

Nevertheless, the feasibility of such an active vision framework remains uncertain: (1) Those planners do not address online domain adaptation, as they rely on offline deep reinforcement learning which requires a huge dataset and time costs; and (2) These planners are not capable of selecting an appropriate goal object but only navigating to a given goal. In particular, the lack of object selection capability is a serious issue. This is because naively visiting every distant object requires a large amount of sense/plan/action cost that is proportional to the number of objects and the robot-to-object distance. Unfortunately, it is a non-trivial task to select an appropriate goal object among distant objects that are typically recognized uncertainly [10]. Note that it is not even clear whether we can do better than a naive action planner that randomly samples one out of the distant objects [11].

To address this issue, we present an efficient simulator for feasibility testing, intending to accelerate the early-stage R&D cycles (e.g., prototyping, training, testing, evaluation). The proposed simulator is designed to identify the degree of difficulty that a robot vision system (sen-

[1]K. Terashima, K. Tanaka, R. Yamamoto, and J. Tay Yu Liang are with Robotics course, Faculty of Engineering, University of Fukui, Japan. {ha209059, tnkknj, mf220362, mf228029}@g.u-fukui.ac.jp

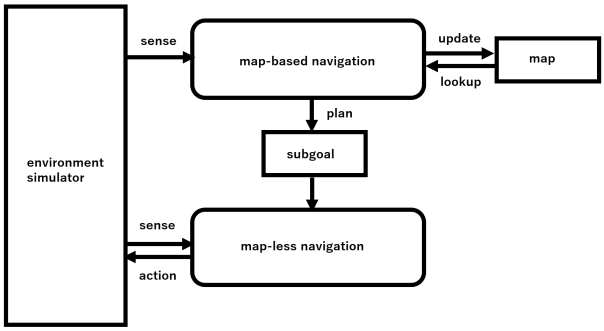

Fig. 2. Hierarchical planner.

sors/recognizers/planners/actuators) would face when applied to a given environment (workspace/objects). This simulator requires a very small amount of real-world data, only one journey per distant object, to function, making it suitable for an efficient R&D cycle. It exploits the analogy between our selection problem "Which object-specific journey should the robot select next?" and the traditional multi-armed bandit problem "Which arm should the player select next?".

Another contribution of this work is that we introduce a lightweight map-based planner on top of mapless navigation, which constitutes a hierarchical planner (Fig.2) that combines the advantages of the map-based planner (i.e., domain-adaptability, goal selection capability) and the mapless planner (i.e., domain-invariance, goal navigation). The effectiveness of the proposed framework has been experimentally verified using the semantically non-trivial scenario "sofa as bookshelf".

A. Discussions

The feasibility test addressed by our simulator aims to identify the degree of difficulty that an active vision system (sensors/recognizers/planners/actuators) would face when deployed in a given environment (workspace/objects). For example, a test result showing that an active vision system has the same level of performance as a naive random action planner is an indicator of the fact that the application is too difficult for that system. As a practical example, in the experimental section of this paper (Section VI), we present the results of a feasibility test on the real-world system we are developing, where the system significantly outperforms a random action planner.

Our simulator requires very little real-world data, only one mapless journey per distant object, to function. This is in contrast to the typical sim-to-real approach (addressed also in our parallel work [12]) that requires rich real-world data from the target workspace for reconstructing the simulated environment. Furthermore, this simulator does not require detailed geometric models of the workspace, which are not available in mapless navigation.

II. Related Work

The problem of image change detection has been studied in various formulations. One popular formulation is a setup of pairwise image comparison for remote sensing applications [3], in which detecting changes across different domains (e.g., across seasons) based on comparisons between query and map images is a topic of ongoing research [4]. However, these studies rely on simplified vision setups such as parallel projection views. This is not the case for ground-view change detection, where the nonlinear perspective projection further complicates the problem. This complex and ill-posed scenario has recently been investigated with support from high-capacity deep learning models [13]. This line of research can be further categorized into 2D-3D comparison [14], 3D-3D comparison [15], and 2D-2D comparison [13], depending on whether the input scene or map model used for comparison is 2D (e.g., monocular RGB image) or 3D (e.g., LIDAR point cloud). In particular, the 2D-2D comparison is one of the most challenging formulations, where the 3D model is not available in both the training and test stages. Our problem formulation falls into this category.

Our approach is most closely related to the research field of ground-view object change detection, aiming to maintain a "bag-of-objects" map, which is an unordered collection of objects detected at the time of mapping within the same workspace. In [16], the problem of detecting significant changes such as the appearance and disappearance of portable traffic lights on high-definition HD maps was addressed by a two-step procedure: (1) Rasterization by projecting map elements from the camera's perspective; (2) Extraction of pyramid features with different resolutions from both the rasterized and camera images. This approach is effective for change detection in large-scale, high-resolution HD maps. Compared with these existing studies, our approach has two key novelties. First, rather than assuming passive vision as in most previous works, we consider active vision (e.g., [17]), including the problem of next-best-view action planning, to determine distant object changes. Second, we do not rely on high-definition 3D models but use only a 2D "bag-of-objects" image model.

Mapless navigation has several variants such as point-goal navigation [5], object-goal navigation [7], and vision-language navigation [8]. Among them, point goal navigation is most relevant to our problem. Formally, it is a workspace-specific lightweight state-to-action map that aims to map a visual input to a next-best-view action. However, it relies on costly offline deep learning and thus is not available for fast online adaptation. In other words, it is assumed that the traversability of the workspace does not change significantly. Such an assumption is relevant in our indoor mobile robot application, where object changes often occupy only a small portion of the input image, and they do not significantly affect the mapless navigation capability. In this work, we aim to explore the mapless navigation framework from a new perspective of map-based active vision.

III. Dataset

Figure 3 shows an overview of the robot. A front-facing onboard camera is used for change detection and collision avoidance. At the time of writing, the mapless navigation module of our robot is under development, and therefore, in the current experiments, it was run in a human-in-the-loop manner with experimenter assistance. Additionally, an LRF was used as an auxiliary proximity sensor for collision avoidance and for recording the ground-truth viewpoint trajectories.

Figure 4 shows an overview of the "sofa as bookshelf" scenario. Although our framework can be generalized to various types of mapped objects, in this work, we consider a specific experimental scenario in which a book is the only object type, to simplify the experiments and analysis. Thus, books with different colors, textures, and sizes are recorded in the "bag-of-objects" format [18]. Now, the purpose of active change detection here is to identify whether each book is unchanged. Note that in cases where a book turns out to be changed, we need to consider a post-process of classifying the book's change state into "appearance" (e.g., the book was occluded by the newly added objects) or "disappearance" (e.g., the book was removed from the workspace). However, such a change state classification task is out of the scope of the current work and left for our future research.

We observed the active change detection task to be very challenging. First, none of the objects were similar to the target object image, nor were they even correctly recognized as books. Since sofas were habitually used as book storage in this workspace, books were often misrecognized as pillows, which are semantically more compatible than books with sofas. Second, sofas were often not categorized as sofas. One reason might be that chairs, a superordinate category of sofas, are a difficult object class to recognize even with the support of affordance cues [19].

IV. Approach

We formulate the active vision problem as reinforcement learning [20], a standard framework for training an action planner. In contrast to other frameworks like supervised and unsupervised learning, the model (i.e., agent) is trained via interaction with the environment in reinforcement learning. Simulated environments are commonly used in robot navigation tasks to reduce the risk of damaging the robot and the cost of robot-environment interaction. For example, in our parallel work [12], we used a photo-realistic physics simulator Habitat. Such simulators are very useful for training environment-independent action planners. However, as reported in recent publications, a model trained in a simulated workspace may have poor performance when deployed in a very different real-world workspace. One straightforward way to address this issue is to build a 3D photo-realistic simulated environment that is equivalent to the target workspace. However, this requires a rich training set acquired in the target workspace and a high-grade 3D reconstruction technique, which significantly limits its applications.

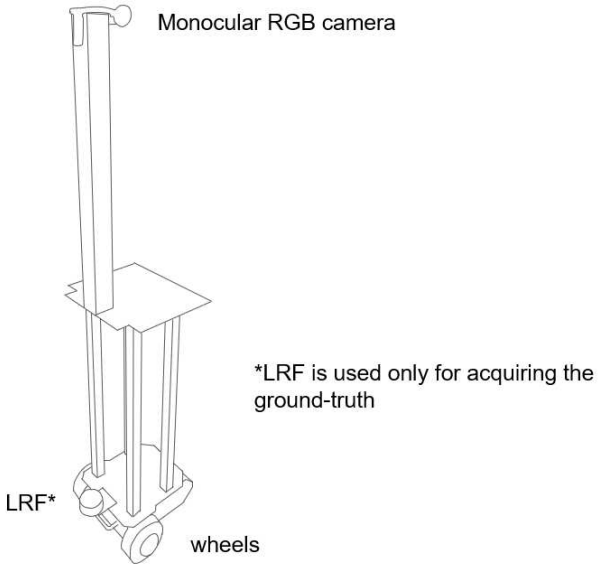

Fig. 3. An active robot vision system.

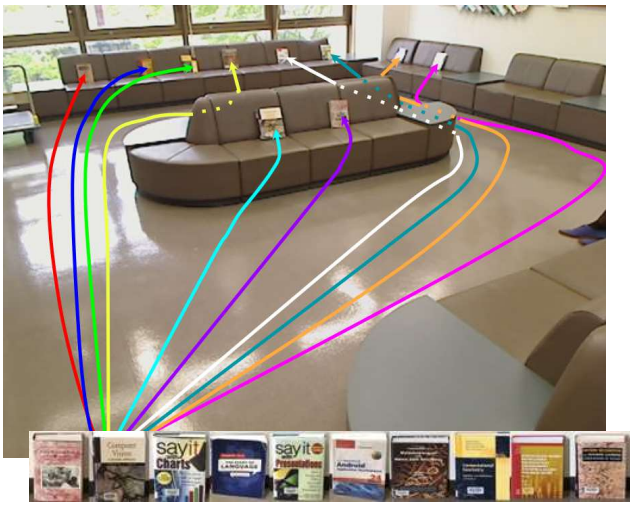

Fig. 4. Viewpoint trajectories from a low-level action planner (top) and map object images (bottom).

Our simulator is introduced to overcome the limitations of these existing simulators. This proposed simulator has several desirable properties. First, it requires only a simple map consisting of N objects that are distantly detected at a certain viewpoint. Second, it does not require rich datasets but only requires one real-world journey per distant object to function. Third, it does not assume the availability of 3D reconstruction technology. This simulator is specifically described as follows. The internal state of the simulator at timestep t is represented by an N-dim state vector $s^t = (s^t[1], \cdots, s^t[N])$, where $s^t[n]$ indicates the unseen part $V_n(s^t[n]), \cdots, V_n(L)$ of each n-th journey V_n at timestep t.

At simulator startup $t = 0$, the state vector is initialized to $s^0 = (1, \cdots, 1)$. At each timestep t, the planner is required to select an appropriate journey n^* ($n^* \in [1, N]$) followed and subsequently sends a "one step forward" action request n^* to the simulator. Then, the simulator returns the first image $V_{n^*}(s^t[n^*])$ in the unseen part of the n^*-th journey, followed by the removal of that image ID from the unseen part (i.e., $s^{t+1}[n^*] \leftarrow s^t[n^*] + 1$). Note that since each object-specific journey is initialized to length L, the robot is not allowed to make more than L action requests per journey. If the top-priority action request by the robot is such an unacceptable action (i.e., $s^t[n^*] = L$), the next-priority action request will be executed instead. In experiments, the number of distant objects was $N = 10$, and each journey consists of a length $L = 100$ image sequence, which constitutes (1) a size NL image set. Figure 5 shows sample images from the journeys. As shown, different journeys often look similar to each other, which may confuse an active vision system.

Our active change detection framework exploits the analogy between the journey selection problem "Which journey should the robot choose next?" and the multi-armed bandit problem "Which arm should the player choose next?." Specifically, the active vision framework is a double-loop algorithm. The outer loop of this algorithm is a map-based navigation phase, while the inner loop is a mapless navigation phase. We focus on the map-based navigation phase, assuming a standard implementation of mapless navigation. One execution of map-based navigation consists of two steps: (1) Evaluate the similarity between each distant object and the target object; (2) Determine "Which distant object should the robot navigate to?" based on the similarity score. In this way, the map-based planner module aims to determine the order by which the distant objects should the robot navigate to.

There are two things to note. First, the exact navigation cost (i.e., sensing, recognition, planning, and action costs) required by an object-specific journey cannot be known in advance, but it can only be estimated by the black-box mapless planner module. In the current experiments, we simply approximate this cost by only the number of image acquisitions (i.e., only the sensing cost). Second, meaningful measurements can only be obtained when the target object is observed closely. In other words, no meaningful measurement can be obtained in cases where other objects are observed or where the target object is observed at a distance. Therefore, a straightforward approach to address this issue would be to navigate the robot toward the most likely object with the highest similarity score at each map-based planning phase.

A. Active Vision

Our active vision approach exploits the analogy between our problem and the "multi-armed bandit (MAB)" problem. In MAB the player is required to select one out of N arms, whereas in our problem a robot is required to select one out of N journeys at each timestep. In MAB the focus of the exploration phase is to learn the expected reward for each arm in MAB, whereas in our problem it is to learn the expectation that the target object will be detected in each journey. In MAB the focus of the exploitation phase is to choose an arm that is expected to maximize the reward, whereas in our problem it is to choose a journey where the expected chance of detecting the target object is maximized.

Fig. 5. Different journeys often look similar to each other, which may confuse the action planner. The panel sequence in each row is samples of images from each journey's raw image sequence, arranged from left to right in ascending order of timestamps.

A simple MAB algorithm that runs the exploration and exploitation phases in sequence is considered. If an agent (i.e., robot) can exploit the current knowledge of the environment, it might be able to identify which part of the environment is worth exploring. However, acquisition of such knowledge requires exploring the environment beforehand.

Therefore, a key hyperparameter L' is introduced to control the balance between the budgets for exploitation and exploration. Given the hyperparameter L', the exploration phase is simply reduced to the task of performing the L' actions in a breadth-first manner for every journey. This will result in obtaining NL' images. On the other hand, the exploitation phase is the task of executing the journeys in a depth-first manner, in an order prioritized by the expected rewards.

At the beginning of the exploitation phase, the top-priority journey with the highest expected reward is selected.

In the exploitation phase, the robot iterates to select the top-priority journey n^* with the highest expected reward

$$n^* = \arg \max_{\{n: s^t[n] < L\}} S(n) \tag{1}$$

among non-completed journeys followed by execution of the selected journey with length $s^t[n^*]$, until the budget is exhausted. Here, $S(\cdot)$ is a given similarity function, one example of which will be illustrated in Section V.

B. Passive Vision

The passive vision module has two roles. One is to keep up-to-date the environmental knowledge: "Which of the distantly detected objects is most likely to be the target object?". This is done by evaluating the similarity of each journey's object to the target object within an image embedding space. Another role is to transfer cues to the action planner that addresses the journey selection problem: "Which of the distantly detected objects should the robot navigate to?". Detailed implementation issues will be discussed in the next section.

V. Implementation

Our approach is implemented in a reinforcement learning framework. We leave the detailed theory and terminology of reinforcement learning to literature (e.g., [20]), and focus on clarifying the component correspondence between reinforcement learning and our approach.

The action space is a collection of distant object IDs $n \in [1, N]$ each of which corresponds to the action plan "The robot navigates to the n-th distant object" or "The robot selects n-th journey".

The state vector was defined as the number of image acquisitions for each journey: $s^t = (s^t[1], \cdots, s^t[N])$.

The state transition is expressed as

$$s^{t+1}[n] \leftarrow \begin{cases} s^t[n] + 1 & (n = n^*) \\ s^t[n] & (\text{Otherwise}) \end{cases}, \qquad (2)$$

when the robot moves forward on the n^*-th journey at timestep t and subsequently acquires an image $V_{n^*}(s^t[n^*])$ at the new viewpoint. The cost is simply approximated by the sensing cost, 1 per acquired image, independent of the executed action n^*, as mentioned earlier.

The expected reward of a distant object at a certain point in time is evaluated as the degree of similarity between the detected object and the target object image with the best imaging conditions among the previously observed images. In our case, this is given by the formula below.

$$S(n) = \max_{j \in [1, s^t[n]-1]} \cos(C(V_q), C(V_n(j))). \qquad (3)$$

The function $C(\cdot)$ is an image embedding using the convolutional neural network vgg16 [21], and returns the signal vector of the 13th layer given a query image input to vgg16.

VI. Experiments

The performance of the proposed active vision framework has been experimentally evaluated using the dataset and methodology explained in Sections III, IV, and V.

Recalling that in our MAB setup the balance can be controlled by the hyperparameter L'. Increasing L' means giving more weight to exploration, whereas reducing L' means putting more weight on exploitation. We repeated experiments for different settings of the hyperparameter L' to evaluate the influence of L' on the performance.

Top-1 accuracy is used as the main performance index. It is defined as the ratio of successful test samples. A test sample consists of a size $N = 10$ set of length $L = 100$ image sequences randomly sampled from the $N = 10$ real-world image sequences, and (2) a ground-truth object randomly sampled from the N map objects. For a given test sample, the active change detection task is regarded to be successful at a certain time $t \in [1, NL]$ if the distant object or journey n^* ($n^* \in [1, N]$) with the highest score at that time is the same as the ground-truth object. Note that in our dataset, each real-world sequence was longer than $L = 100$ in length, so each random sampling would generate a different sequence.

Figure 6 shows the experimental results. As shown, if the budget paid for the exploration stage NL' is too small (e.g., $L' = 1$), the prior knowledge of the environment in the exploitation stage is often insufficient and the performance degrades to the level of a naive random action planner. In contrast, if the budget for exploration is too large (e.g., $L' = 50$), the performance of the exploitation stage could be maximized, but at the expense of increasing cost.

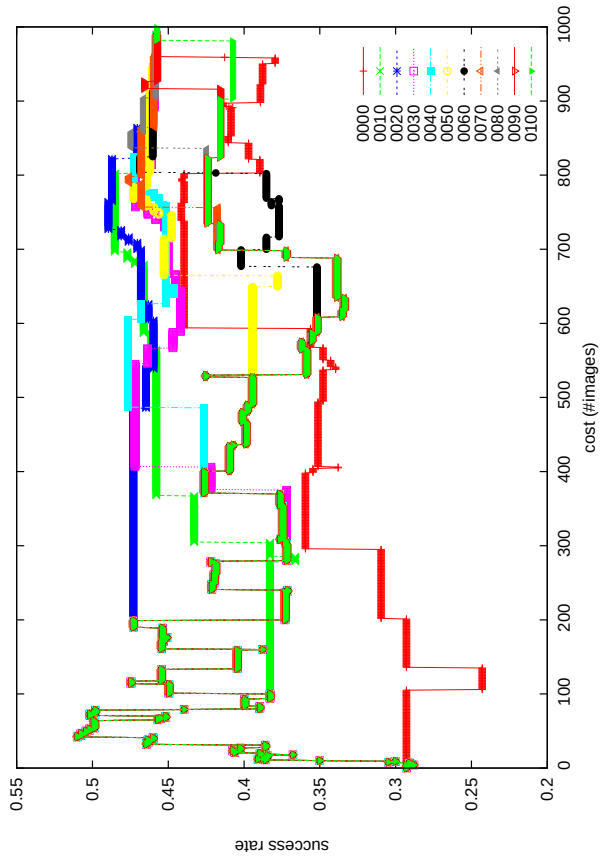

Fig. 6. Performance results. Vertical axis: detection accuracy. Horizontal axis: navigation cost. Legend: hyperparameter L'.

We conducted ablation research considering the diversity of available budgets. In general, the available budget depends on the application/user. For example, for a user robot whose primary mission is mail delivery, change detection is often a secondary task that cannot be paid much budget. In contrast, for a user robot whose main mission is map maintenance, a large amount of budget may be available for change detection. Considering such diversity in available budgets, we performed performance evaluations for different settings of available budgets. Specifically, we simulated 999 different cases, ranging from $L' = 1$ to $L' = 999$, and examined the user's performance for the available budget. The results are summarized in Fig. 7. As shown, users characterized by $L' = 20$ benefited the most from the active vision algorithm considered here. It is also noteworthy that the proposed planner proved to be significantly superior to a naive random action planner in the experiments considered here.

In conclusion, the feasibility tests considered here demonstrate that our active vision system has a sufficient level of recognition ability to select an appropriate target object

among distant objects uncertainly recognized. Nevertheless, it remains unclear whether this result can be generalized to other situations that may be encountered in the workspace. For example, the performance can vary depending on various factors, such as the robot-to-object distances, the visual object recognizability, and visual foreground-background compatibility. Nevertheless, we believe that this test still serves as an efficient way to test, for a given recognizer-planner pair, whether that recognizer can provide useful cues to the planner.

VII. Conclusions

We explored a challenging action planning problem in a distant object change detection scenario: "Which object-specific journey should the robot navigate next?". We build a map-based planner module for goal object selection on top of the goal-oriented mapless navigation module, which constitutes a hierarchical planner. We also introduced an efficient simulator for a feasibility test, for a given distant object recognizer, to decide whether the recognizer can provide useful cues for an action planner. We experimentally verified the effectiveness of the proposed method using a semantically non-trivial dataset "sofa as bookshelf."

The proposed simulator extremely simplifies the lower-level planning tasks, which significantly accelerates the sense-plan-act cycle in the high-level planning task. As an additional advantage, it requires only one real-world journey experience per distant object to function, which is a particularly property for the physical robot design stage.

There are several avenues for future research. One of the avenues is to improve the proposed multi-armed bandit framework and improve its performance. Empirical verification can be performed by conducting evaluation experiments and case studies in application scenarios where distant object detection is critical (e.g., map maintenance applications). Another interesting area for study is investigating the factors that determine why and how map-based and mapless planners collaborate [9] to improve overall performance. Finally, research on the effectiveness of specific strategies to transfer the findings in the proposed lightweight planner to more sophisticated but expensive planners (e.g., deep reinforcement learning planners [22]) would be an important direction of future research.

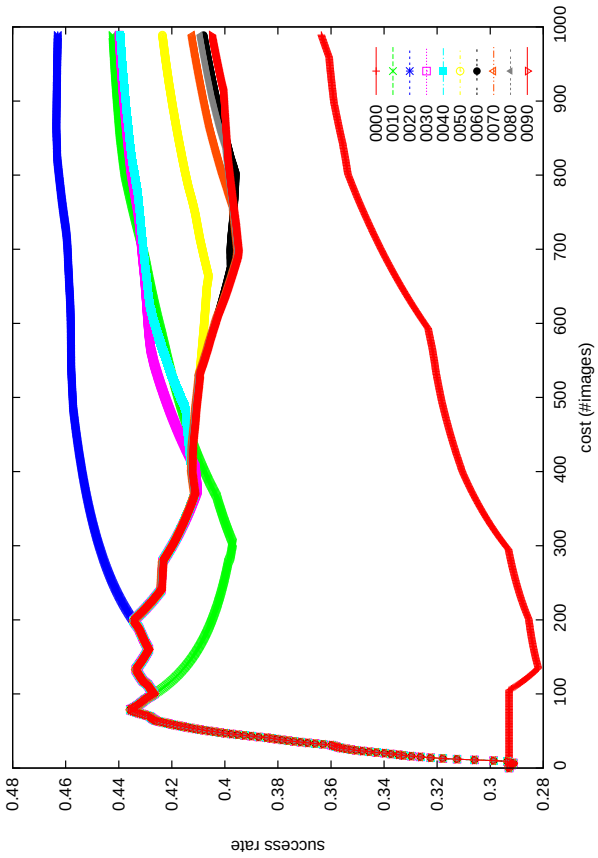

Fig. 7. Average performance versus available budget.

References

[1] Bo Liu, Xuesu Xiao, and Peter Stone. A lifelong learning approach to mobile robot navigation. *IEEE Robotics and Automation Letters*, 6(2):1090–1096, 2021.

[2] Aisha Walcott-Bryant, Michael Kaess, Hordur Johannsson, and John J Leonard. Dynamic pose graph slam: Long-term mapping in low dynamic environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1871–1878. IEEE, 2012.

[3] Fenlong Jiang, Maoguo Gong, Tao Zhan, and Xiaolong Fan. A semisupervised gan-based multiple change detection framework in multi-spectral images. *IEEE Geoscience and Remote Sensing Letters*, 17(7):1223–1227, 2019.

[4] Aparna Taneja, Luca Ballan, and Marc Pollefeys. Geometric change detection in urban environments using images. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2193–2206, 2015.

[5] Joel Ye, Dhruv Batra, Erik Wijmans, and Abhishek Das. Auxiliary tasks speed up learning point goal navigation. In *Conference on Robot Learning*, pages 498–516. PMLR, 2021.

[6] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023.

[7] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020.

[8] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 8455–8464, 2021.

[9] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam, 2020.

[10] Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2023.

[11] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997.

[12] Mitsuki Yoshida, Kanji Tanaka, Ryogo Yamamoto, and Daiki Iwata. Active semantic localization with graph neural embedding. In *2023 7th Asian Conference on Pattern Recognition, ACPR 2023*, 2023.

[13] Ken Sakurada, Takayuki Okatani, and Koichiro Deguchi. Detecting changes in 3d structure of a scene from multi-view images captured by a vehicle-mounted camera. In *2013 IEEE Conference on Computer*

Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013, pages 137–144. IEEE Computer Society, 2013.

[14] Emanuele Palazzolo and Cyrill Stachniss. Fast image-based geometric change detection given a 3d model. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 6308–6315. IEEE, 2018.

[15] Mani Golparvar Fard, Feniosky Peña-Mora, and Silvio Savarese. Monitoring changes of 3d building elements from unordered photo collections. In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*, pages 249–256. IEEE Computer Society, 2011.

[16] Lei He, Shengjie Jiang, Xiaoqing Liang, Ning Wang, and Shiyu Song. Diff-net: Image feature difference based high-definition map change detection for autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2635–2641. IEEE, 2022.

[17] Kanji Tanaka, Hongbin Zha, and Tsutomu Hasegawa. Viewpoint planning in map updating task for improving utility of a map. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 1, pages 729–734. IEEE, 2003.

[18] Yang Cao, Changhu Wang, Zhiwei Li, Liqing Zhang, and Lei Zhang. Spatial-bag-of-features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3352–3359. IEEE, 2010.

[19] Helmut Grabner, Juergen Gall, and Luc Van Gool. What makes a chair a chair? In *CVPR 2011*, pages 1529–1536. IEEE, 2011.

[20] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] Haiyan Yin and Sinno Pan. Knowledge transfer for deep reinforcement learning with hierarchical experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.