Active Map-Matching: Active Self-localization by VPR-to-NBV Knowledge Transfer

Tanaka Kanji

***Abstract*— Training a next-best-view (NBV) planner for active cross-domain self-localization is an important and challenging problem. Unlike typical in-domain settings, the planner can no longer assume the environment state being constant, but must treat it as a high-dimensional component of the state variable. This study is motivated by the ability of recent visual place recognition (VPR) techniques to recognize such a high-dimensional environment state in the presense of domain-shifts. Thus, we wish to transfer the state recognition ability from VPR to NBV. However, such a VPR-to-NBV knowledge transfer is a non-trivial issue for which no known solution exists. Here, we propose to use a reciprocal rank feature, derived from the field of transfer learning, as the dark knowledge to transfer. Specifically, our approach is based on the following two observations: (1) The environment state can be compactly represented by a local map descriptor, which is compatibile with typical input formats (e.g., image, point cloud, graph) of VPR systems, and (2) An arbitrary VPR system (e.g., Bayes filter, image retrieval, deep neural network) can be modeled as a ranking function. Experiments with nearest neighbor Q-learning (NNQL) show that our approach can obtain a practical NBV planner even under severe domain-shifts.**

I. Introduction

In this paper, we aim to train a next-best-view (NBV) planner for active cross-domain self-localization. Given a landmark map built in a past domain (e.g., weather, season, times of the day), the goal of self-localization is to localize the robot-itself, using relative measurements from on-board landmark sensor and odometory [1]–[3]. This cross-domain self-localization problem becomes a challenging one due to appearance/removal of landmarks as well as perceptual aliasing. Thus far, most of previous works on cross-domain self-localization suppose a passive observer (i.e., robot) and do not take into account the viewpoint planning or controlling the observer [4]–[6]. However, such a passive self-localization setting can be ill-posed, due to spatial sparseness of domain-invariant landmarks (e.g., pole-like landmarks [7]), salient viewpoints, and sufficient landmark views (Fig. 1). Therefore, we consider active self-localization with an active observer that can adapt its viewpoint trajectory, avoiding non-salient scenes that may not provide sufficient landmark view, or moving efficiently towards places which are most informative, in the sense of reducing the sensing and computation costs.

Our work has been supported in part by JSPS KAKENHI Grant-in-Aid for Scientific Research (C) 17K00361, and (C) 20K12008.

K. Tanaka is with Faculty of Engineering, University of Fukui, Japan. tnkknj@u-fukui.ac.jp

We would like to express our sincere gratitude to Kanya Kurauchi for development of deep learning architecture, and initial investigation on deep reinforcement learning on the dataset, which helped us to focus on our VPR-to-NBV project.

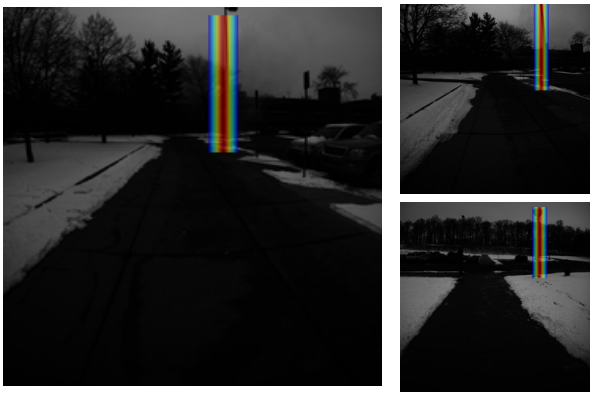


Fig. 1. Motivation of the proposed work. Domain-invariant landmarks are spatially sparsely distributed, which makes the cross-domain self-localization an ill-posed problem for a passive observer, and leads to the necessity of an active self-localization framework. Shown in the figure are domain-invariant pole-like landmarks, which are extracted by state-of-the-art deep holistic landmark detection in [8].

Training an NBV planner for active cross-domain self-localization is computationally very challenging, and most likely the reason why there is little solution to this problem. Unlike in-domain self-localization, the planner can no longer assume the environment state being constant, but must treat it as a high-dimensional component of the state variable. This significantlly increases the dimensionality and size of the state space, which makes a direct implementation of standard training procedure such as reinforcement learning computationally intractable.

This study is motivated by the ability of recent visual place recognition (VPR) techniques to recognize such a high-dimensional environment state in the presense of domain-shifts [9]–[11]. Thus, we wish to transfer the state recognition ability from VPR to NBV. However, such a VPR-to-NBV knowledge transfer is a non trivial issue for which no known solution exists.

Here, we propose a novel VPR-to-NBV knowledge transfer framework that uses reciprocal rank feature [12] as the dark knowledge. Specifically, our approach is based on the following two observations: (1) The environment state can be compactly represented by a local map descriptor, which is compatibile with typical input formats (e.g., image, point cloud, graph) of VPR systems [13], and (2) An arbitrary VPR system (e.g., Bayes filter [1], image retrieval [2], deep neural network [12]) can be modeled as a ranking function. The proposed approach is inspired by the recent development of transfer learning, where rank matching is used as the dark knowledge to transfer from a teacher to a student model (e.g.,

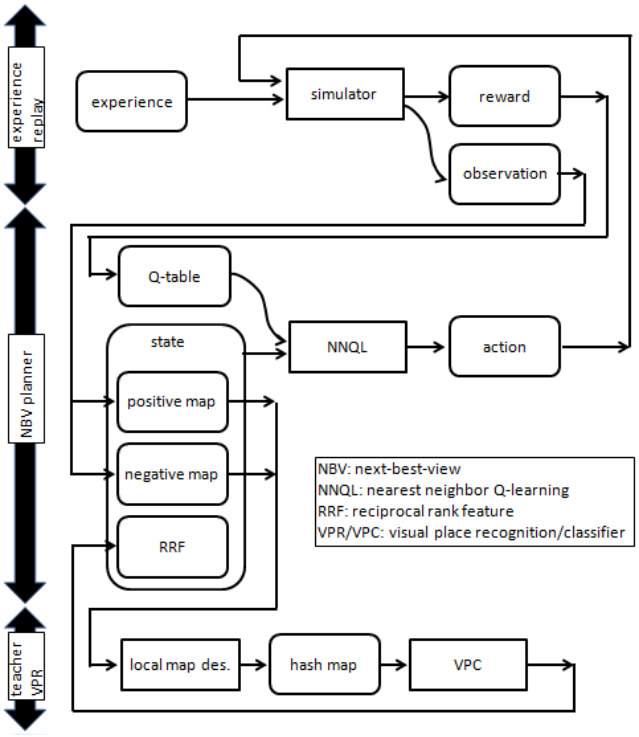


Fig. 2. Algorithm pipeline. In this paper, we focus on the module for knowledge transfer from VPR to NBV.

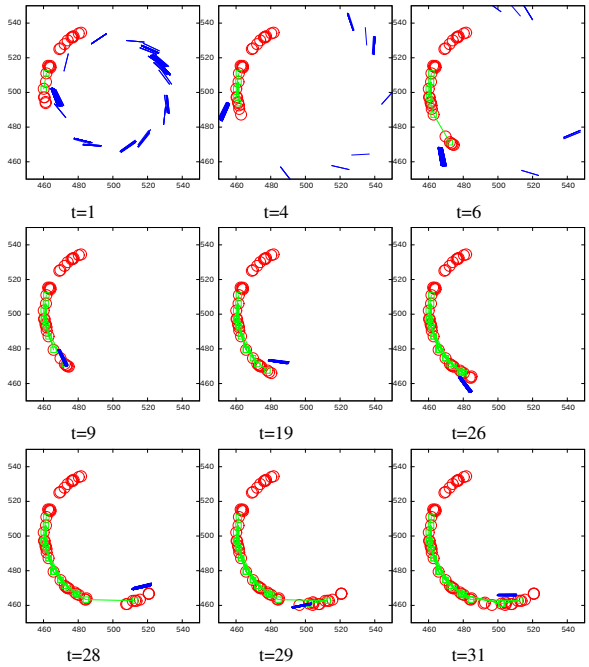


Fig. 3. Examples of landmark detection and self-localization results. 'o': detected landmark. '-': hypothesized viewpoint location and orientation, for highly-ranked hypotheses that receive score values higher than 50%.

classifier) [14]. Such a ranking -based knowledge transfer has been further explored in our previous studies on knowledge distillation [15], rank fusion [16], image change detection [17], and graph convolutional neural network [12]. While these existing studies focus on the passive self-localization setting, the current study explores the issue of active self-localization. The proposed framework has the following advantages. (1) Compact: While a local map can grow in unbounded fashion as the robot explores a large area, it can be compressed into a compact bit pattern by using the hash-map technique [18]. (2) Discriminative: The reciprocal rank feature can serve as a discriminative dark knowledge to differentiate between different states [12]. (3) Drift-free: Viewpoint-drifts between local maps can be suppressed by using the local map descriptor [13]. Experiments with nearest neighbor Q-learning (NNQL) [19] show that our approach can obtain a practical NBV planner even under severe domain-shifts.

II. Approach

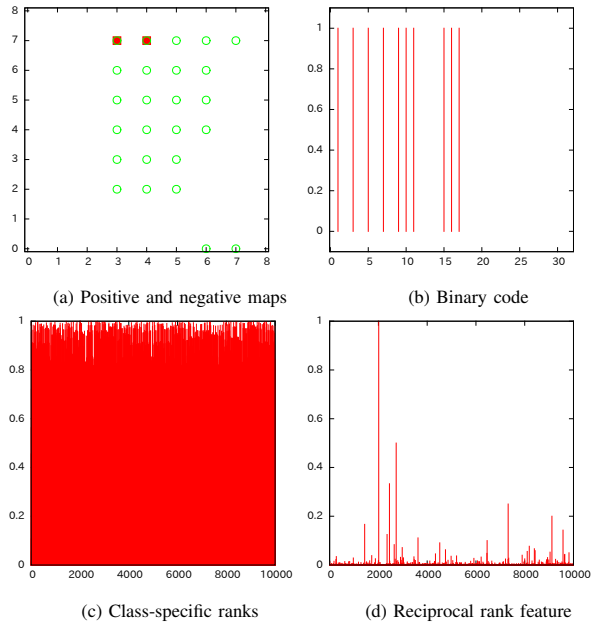
Figure 2 depicts the architecture of the proposed framework. Without losing generality, we consider a typical 2D robot navigation scenario [20] (Fig. 3), which is character-sized by 2D point landmarks (e.g., pole-like landmarks) in a 2D bird's-eye-view environment, and by 2D robot poses (x, y, θ) and actions (forward, rotate) on the moving plane.

A 2D omni-directional range finder (ORF) equipped on the robot provides both positive and negative observations for each direction α ($|\alpha| \leq \pi$) (Fig. 4). Positive observations are the 2D end-points of ORF readings $\{d(\alpha)\}$ that are associated to obstacles or landmarks. Negative observations are the areas swept by ORF readings before reaching the end-points that are associated to free-space.

The local map descriptor incorporates both positive and negative observations. These positive and negative maps are represented respectively with an array of 2D relative landmark locations and with a local 2D regular grid map where each grid cell can take either one of the states: unknown, free, or occupied (Fig. 4).

The training engine for the NBV planner adapts the recently developed framework of NNQL [19]. NNQL has the following desirable properties. (1) Unlike many exisiting Q-learning frameworks, NNQL provides a good approximation of the Q-values for even unexperienced states and/or actions which the robot has not been explored yet. (2) Therefore, NNQL is expected to be robust against domain gaps between training and test states. (3) Unlike many Q-learning algorihtms, the convergence of NNQL is guaranteed. (4) Its convergence speed could be further boosted by employing an external NN retrieval engine.

A VPR system is generally modeled as a ranking function, which can work with arbitrary VPR systems (e.g., Bayes filter [1], image retrieval [6], deep neural network [21]). It evaluates the likelihood of the robot being located at each predefined place class, given a query scene. In this study, we address a specific scenario, where a scene is represented by the local map descriptor, and thus each place class is predefined as a cluster of descriptors. For the clustering, we use the standard clustering algorithm of k-means on



(a) Positive and negative maps (b) Binary code

(c) Class-specific ranks (d) Reciprocal rank feature

Fig. 4. A visualization of intermadiate results. (a) An example of the positive and negative maps in the local map descriptor. The '□' and '○' indicate the locations of positive and negative grid cells. (b) Binary code vector. (c,d) Feature vectors for the class-specific rank and reciprocal rank features. It can be seen that the latter vector is much discriminative than the former one.

a training set, although the problem of place clustering to optimize pureness of place clusters is an important topic of on-going researches [22].

We observe that domain-invariant landmarks are often spatially very sparse. For example, domain-invariant pole-like landmarks in [7] (Fig. 1) are significantlly more spatially-sparse than general purpose landmarks such as local keypoint descriptors [23]. Such a characteristics will be also taken into consideration in the proposed approach.

A. Active Self-localization Problem

The active self-localization task takes a motion observation history s_t at a time step t and determines the NBV action a_t, which consists of a 2D translation (x[m], y[m]) and a rotation θ[deg]. The action space is defined as the grid of $\{(x,y)|\ |x|+|y|>0,\ x,y\in\{0,\pm10,\pm20\}\}$. The state space is defined as the space of high-dimensional local map descriptor.

B. Reinforcement Learning

Reinforcement learning (RL) [24] is used to train the NBV planner. In the active self-localization application, a model-based planner such as probabilistic road map [25] is not applicable, because of the unavailability of world-centric map information. Instead, a model-free planner is preferred, to map the available ego-centric map information to the world-centric NBV action plan. RL is a standard approach to such a model-free planner.

In RL, the reward function rewards a given state-action in terms of the total self-localization performance. A naive strategy is to define an estimate of the self-localization performance (i.e., mAP) directly as a reward function. Unfortunately, such an estimation is computationally expensive, and the training cost per episode grows in proportion to the trajectory length.

To address this issue, in our approach, the number of detected landmarks is viewed as a light-weight proxy for such an expensive reward function. This strategy is based on our observation that the number of detected landmarks has strong positive correlation with the self-localization performance (Fig. 1) [8]. Thanks to such a lightweight proxy, the overhead of the reward function against the total training cost becomes negligively small.

C. Local Map Descriptor

Viewpoint drift between training and test scenes is a major source of difficulty in viewpoint planning [26]. It is inherently caused by accumulative pose tracking errors during long-term navigation, as well as robot kidnapped problems. A straightforward way to address this issue is aligning every pair of training and test scenes (e.g., via scan/map -matching [27]) in online. Unfortunately, this naive strategy requires a significant increase in time cost per episode.

Here, we propose to pre-align every training/test scene's coordinate with a domain-invariant coordinate system (ICS), to avoid the need of online alignment. While the problem of finding such an ICS (i.e., an origin and axes) for a given scene is in general ill-posed [13], fortunately in our application domain the location of a domain-invariant landmark can be used as the ICS's origin (Fig. 1) [28], and moreover, the spatial distribution of domain-invariant landmarks can be used as a cue to determine the ICS's orientation (Fig. 3).

The detailed procedure for determining ICS is as follows. First, the ICS's axes are determined so that the Entropy of landmark locations along the x-axis becomes maximum [29]. Then, the ICS's origin is fine-adjusted to the location of the landmark with the shortest distance to the robot's viewpoint. This fine-adjustment is triggered only when the shortest distance does not exceed a predefined threshold $T_d = 1$ [m].

It should be noted that ICS is effective both for VPR and NBV, as demonstrated in our previous studies [28] and [8]. In VPR, the ICS provides a viewpoint-invariant discriminative scene descriptor [28]. In NBV, the ICS suppresses the effects of viewpoint-drifts between training and test scenes [8].

D. Hash Map

The hash-map technique in [16] is employed to compress the variable-size local map into a constant-length binary code. In general, a local map can grow in unbounded fasion as the robot explores a large area. With a few exceptions (e.g., [30]), most existing VPR algortihms can not address such a variable-size input format. To address this issue, we map the variable size local map into a constant-length Z-dim representation z ($Z = 32$). For this, the hash-map is used with a random projection $Y = PX$, followed by a mod operation $z[j] = \sum_{j\in S_Z} y[i]$ where $S_Z = \{j|(i \bmod Z) = j\}$.

More specifically, the positive and negative information in the local map are first translated to two separete grid

maps, named positive and negative maps, either of which is represented by a 8×8 grid with spatial resolution of 1[m]. Then, either map is compressed via the above hash-map into a Z-dim representation. Then, the results are concatenated and binarized to a $(2Z)$-dim bit-code.

It should be noted that the projection matrix P can be recovered on-the-fly as an array of pseudo random numbers given a predefined random seed. Thus, the space cost for the projection matrix is constant and negligibly low.

E. Reciprocal Rank Features

The reciprocal rank feature [12] is used as the dark knowledge for the VPR-to-NBV knowledge transfer. This strategy is based on the observation that any off-the-shelf VPR system can be modeled as a ranking function that ranks a set of predefined place classes based on their similarity to a given query scene. In this study, a query scene is representated by a local map descriptor, and the place classes are defined in offline by clustering training descriptors into 10,000 place classes via k-means clustering. The proposed approach has several desirable properties. First, using rank values as the dark knowledge is theoretically supported by the rank matching loss, which is used in recent transfer learning techniques such as knowledge distillation [14]. Second, the reciprocal rank is additive feature [31]. Third, the reciprocal rank values are successfully used in multi-modal information retrieval [32] and in VPR [16].

F. Nearest Neighbor Q-Learning (NNQL)

A key difference of NNQL [19] from the standard QL is that the Q-value of an input state-action-pair (s,a) is approximated by a collection of Q-values that are associated with its nearest neighbors $N(s,a)$:

$$Q(s,a) = \frac{1}{|N(s,a)|} \sum_{(s',a')\in N(s,a)} Q(s',a') \tag{1}$$

We build a set of independent $|A|$ NN engines for individual $|A|$ action candidates. Thus, the Q-function is approximated by:

$$Q(s,a) = \arg\max_{a} \frac{1}{|N(s|a)|} \sum_{(s',a)\in N(s|a)} Q(s',a), \tag{2}$$

where $N(s|a)$ is the nearest neighbors of (s,a) conditioned on a given action a. In our implementation, the set $N(s|a)$ is defined as $\{(s',a')|\ |s'-s|\leq 2, a'=a\}$, where $|\cdot|$ is L1-norm. Such an action-specific NNQL also can be viewed as an instance of RL [24].

III. Experimental Results

The goal in this experiment is to test the effectiveness of the proposed framework, whose scene and VPR models respectively are based on the local map descriptor (LMD) and reciprocal rank transfer (RRT). We generated a collection of 3,300 different settings, each of which consists of the robot initial viewpoint and a configuration of domain-invariant landmarks. To make active self-localization a non-trivial problem, we focus on challenging environments, which consist of many repetitive/symmetric structures, and a few discriminative structures. Specifically, we consider three different types of landmark configuration: "CIRCULAR", "ROAD", and "RECTANGULAR" (Fig. 5). The "CIRCULAR" configuration is intended for an application in which poles lined up along the perimeter of a circular park are used as landmarks. The "ROAD" configuration envisions an application in which pole landmarks are lined up at random intervals in [0, 1.0] [m] along a pole-lined road. The "RECTANGULAR" configuration envisions an application where pole landmarks are lined up along the perimeter of a square parking lot. The training and test environments are created in three steps. (1) First, prototypes of the "CIRCULAR", "ROAD" and "RECTANGULAR" landmark configurations are created in the following manner. For "CIRCULAR" configuration, each i-th landmark location (x_i,y_i) is determined by: $[x_0+R\cos(\Delta(\theta)r2\pi/N),\ y_0+R\sin(\Delta(\theta)r2\pi/N)]$. r is a sample from a uniform distribution in [0, 1]. For "ROAD" configuration, the first landmark's location is determined $(x_0,y_0)=(0,500)$ and then each i-th landmark's location $(i>1)$ is incrementallly generated at the location $(x_i,500)=(x_{i-1}+r\Delta L,500)$, until $x_i>1000$. For "RECTANGULAR" configuration, landmarks are generated in the similar incremental manner as in "ROAD", but along the perimeter of a square $[400,600]\times[400,600]$ instead of along the single line segment. $\Delta L=1.0[m]$. N=100. $R=40$m. (2) Then, a set of 1,000 training environments are created by randomly modifying each prototype landmark configuration. (3) Then, a set of 100 test environments are created by randomly modifying each of 100 randomly-selected training environments for M times, where M is set to 10% of the number of landmarks in the environment of interest. For the random modification of an environment, we use two kinds of operations: appearance and removal of landmarks. The appearance operation adds a new landmark at a random location. The removal operation removes a randomly-chosen one of the existing landmarks. It should be noted that the above procedure yields a challenging set of training/test environments that consist of near-duplicate, repetitive and symmetric structures, which suffers from perceptual aliasing. The on-board ORF has the azimuth resolution of 1.0 deg and its maximum range is 10.0 m. In the training and test stages, the number of actions per episode is set to 100.

Difficulty of the landmark-detection and self-localization tasks strongly depends on the distance from the initial robot viewpoint to the closest landmark. If the distance is larger than the ORF's maximum range, a robot's exploring behavior such as random walk would be required to encounter the first landmark. To avoid dependency of the initial setting, each training/test episode starts with such an initial viewpoint at which at least one landmark exists in the field-of-view. Likewise, action candidates are restricted to those by which the robot moves to a viewpoint at which at least one already-detected-landmark exists within the field-of-view. During a navigation task, the robot repeats the three steps: observation, plan, and action, as in Section II.

In the test stage, we perform a highly accurate map-

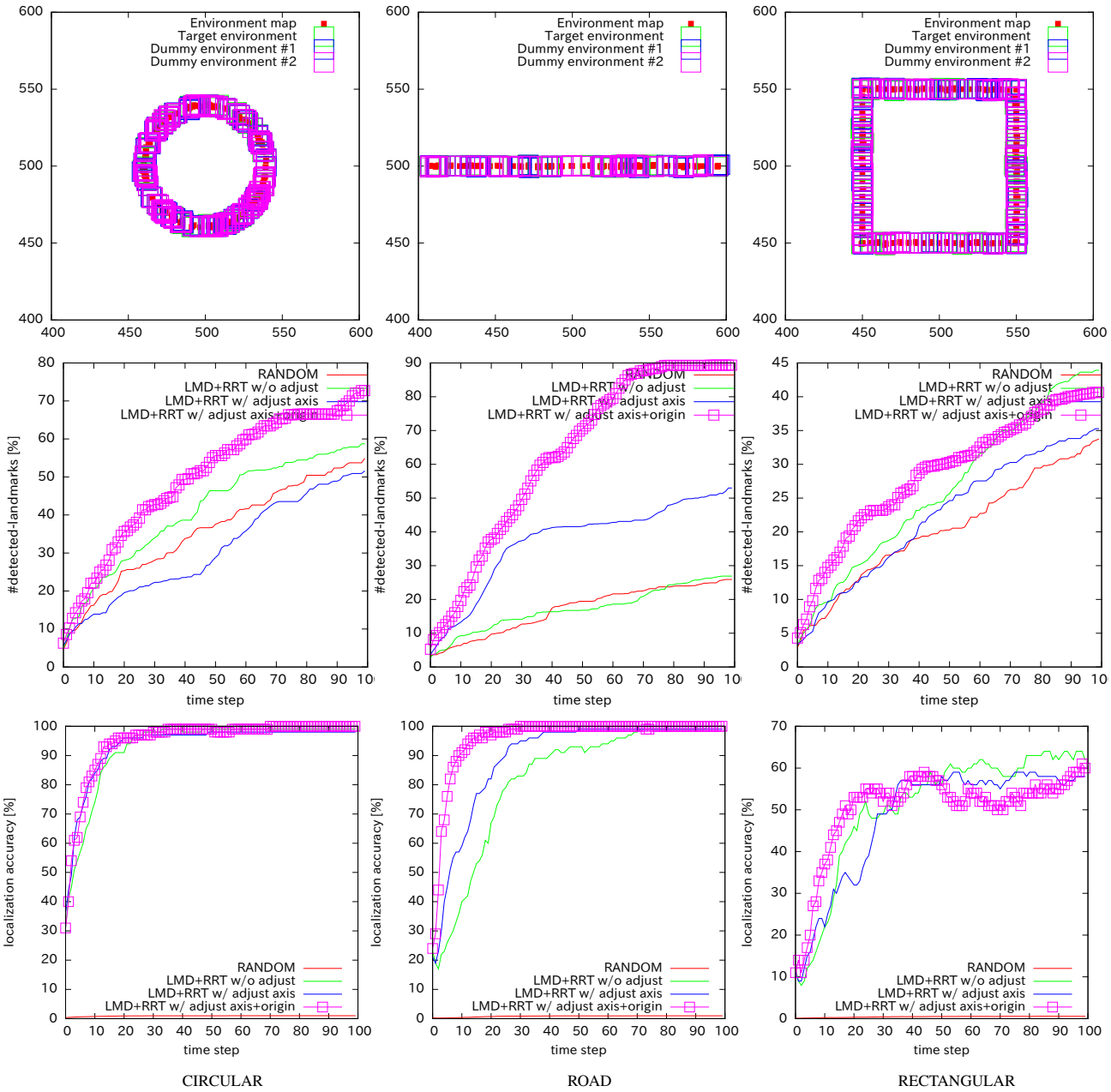


Fig. 5. Experimental results. Each column from left to right shows "CIRCULAR", "ROAD", and "RECTANGULAR". For the sake of clarity, only a portion of the environment is shown for the "ROAD" type configuration. Top: Examples of the traininig and test environments. For the test environments, the target environment and two dummy environments are shown. Middle: Landmark detection performance versus time step. Bottom: Self-localization performance versus time step.

matching -based self-localization, at each time-step during the robot navigation, because there is no severe restriction on the computational cost per episode as in the training stage. For the map-matching -based self-localization, we employ RANSAC map-matching between the latest ego-centric local map and the a-priori given world-centric map. More specifically, a 2-point RANSAC algorithm is used to hypothesize and score a set of 3-dof viewpoint hypotheses. For performance evaluation, the success of the self-localizatiojn is deteremined by whether the error with respect to the ground-truth is less than 1m.

For performance comparsion, we developed four different NBV methods: "RANDOM", "LMD+RRT w/o adjust", "LMD+RRT w/ adjust axis", and "LMD+RRT w/ adjust origin+axis". "RANDOM" is a naive strategy that randomly selects an action from the executable action candidates at each time step. "LMD+RRT w/ adjust origin+axis" is the proposed strategy, which uses LMD for environment state represenation, RRT for VPR-to-NBV knowledge transfer, and ICS for aligning origin/axes of the ego-centric local map. The methods "LMD+RRT w/ adjust axis" and "LMD+RRT w/o adjust" are ablations of the proposed method. The former

does not use the alignment of the origin. The latter does not use both the alignment of the origin and axes.

Figure 3 demonstrates typical viewpoint trajectories for the proposed NBV planner at the test stage. As can be seen, the viewpoint hypotheses are gradually refined as the robot moves and detects new landmarks.

Figure 5 shows the basic performance of methods. It can be seen that the proposed method, which combines LMD and RRT with adjusting the axes and origin, outperforms all the other methods in both landmark detection and active self-localization tasks.

IV. Conclusions

In this study, we explored the novel task of active cross-domain self-localization from the perspective of VPR-to-NBV knowledge transfer. Specifically, our approach is based on the following two observations: (1) The environment state can be compactly represented by a local map descriptor, which is compatibile with typical input formats (e.g., image, point cloud, graph) of VPR systems, and (2) An arbitrary VPR system (e.g., Bayes filter, image retrieval, deep neural network) can be modeled as a ranking function. In our contributions, we proposed to use a reciprocal rank feature, derived from the field of transfer learning, as the dark knowledge to transfer. Experiments showed that our approach can obtain a practical NBV planner even under severe domain-shifts.

References

[1] M. Himstedt and E. Maehle, "Semantic monte-carlo localization in changing environments using rgb-d cameras," in *2017 European Conference on Mobile Robots (ECMR)*. IEEE, 2017, pp. 1–8.

[2] H. Gao, X. Zhang, J. Yuan, J. Song, and Y. Fang, "A novel global localization approach based on structural unit encoding and multiple hypothesis tracking," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 11, pp. 4427–4442, 2019.

[3] J. Neira, J. D. Tardós, and J. A. Castellanos, "Linear time vehicle relocation in slam," in *ICRA*. Citeseer, 2003, pp. 427–433.

[4] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE Int. Conf. Robotics and Automation*. IEEE, 2012, pp. 1643–1649.

[5] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," vol. 32, no. 14, 2013, pp. 1645–1661.

[6] E. Garcia-Fidalgo and A. Ortiz, "ibow-lcd: An appearance-based loop-closure detection approach using incremental bags of binary words," *IEEE Robotics and Automation Letters*, pp. 3051–3057, 2018.

[7] A. Schaefer, D. Büscher, J. Vertens, L. Luft, and W. Burgard, "Long-term urban vehicle localization using pole landmarks extracted from 3-d lidar scans," in *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 2019, pp. 1–7.

[8] K. Tanaka, "Domain-invariant nbv planner for active cross-domain self-localization," *arXiv preprint*, 2021.

[9] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2016.

[10] N. Merrill and G. Huang, "CALC2.0: Combining appearance, semantic and geometric information for robust and efficient visual loop closure," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019.

[11] J. Spencer, R. Bowden, and S. Hadfield, "Same features, different day: Weakly supervised feature learning for seasonal invariance," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2020, pp. 6459–6468.

[12] K. Takeda and K. Tanaka, "Boosting self-localization with graph convolutional neural networks," *Visapp*, 2021.

[13] Y. Takahashi, K. Tanaka, and N. Yang, "Scalable change detection from 3d point cloud maps: Invariant map coordinate for joint viewpoint-change localization," in *2018 21st Int. Conf. Intelligent Transportation Systems*. IEEE, 2018, pp. 1115–1121.

[14] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[15] T. Hiroki and K. Tanaka, "Long-term knowledge distillation of visual place classifiers," in *2019 IEEE Intelligent Transportation Systems Conference*. IEEE, 2019, pp. 541–546.

[16] K. Tanaka, "Unsupervised part-based scene modeling for visual robot localization," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6359–6365.

[17] ——, "Detection-by-localization: Maintenance-free change object detector," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4348–4355.

[18] K. Saeki, K. Tanaka, and T. Ueda, "Lsh-ransac: An incremental scheme for scalable localization," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3523–3530.

[19] D. Shah and Q. Xie, "Q-learning with nearest neighbors," *arXiv preprint arXiv:1802.03900*, 2018.

[20] T. Bailey, E. M. Nebot, J. Rosenblatt, and H. F. Durrant-Whyte, "Data association for mobile robot navigation: A graph theoretic approach," vol. 3, pp. 2512–2517, 2000.

[21] N. Yang, K. Tanaka, Y. Fang, X. Fei, K. Inagami, and Y. Ishikawa, "Long-term vehicle localization using compressed visual experiences," pp. 2203–2208, 2018.

[22] K. Tanaka, "Self-supervised map-segmentation by mining minimal-map-segments," in *IEEE Intelligent Vehicles Symposium (IV)*, 2020.

[23] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int. J. robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[24] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[25] R. Geraerts and M. H. Overmars, "A comparative study of probabilistic roadmap planners," in *Algorithmic foundations of robotics V*. Springer, 2004, pp. 43–57.

[26] B. L. Floriani, N. Palomeras, L. Weihmann, H. Simas, and P. Ridão, "Model-based underwater inspection via viewpoint planning using octomap," in *OCEANS 2017-Anchorage*. IEEE, 2017, pp. 1–8.

[27] B. Zhou, Z. Tang, K. Qian, F. Fang, and X. Ma, "A lidar odometry for outdoor mobile robots using ndt based scan matching in gps-denied environments," in *IEEE Annual Int. Conf. CYBER Technology in Automation, Control, and Intelligent Systems*, 2017, pp. 1230–1235.

[28] R. Yamamoto, K. Tanaka, and K. Takeda, "Invariant spatial information for loop-closure detection," in *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, 2019, pp. 1–6.

[29] S. Olufs and M. Vincze, "Robust single view room structure segmentation in manhattan-like environments from stereo vision," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 5315–5322.

[30] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE Conf. computer vision and pattern recognition*, 2017, pp. 652–660.

[31] G. V. Cormack, C. L. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in *Int. ACM SIGIR Conf. Research and development in information retrieval*, 2009, pp. 758–759.

[32] A. Mourão, F. Martins, and J. Magalhaes, "Multimodal medical information retrieval with unsupervised rank fusion," *Computerized Medical Imaging and Graphics*, vol. 39, pp. 35–45, 2015.