2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

PartSLAM: Unsupervised Part-based Scene Modeling for Fast Succinct Map Matching

Hanada Shogo Tanaka Kanji

Abstract— **In this paper, we explore the challenging 1-to-N map matching problem, which exploits a compact description of map data, to improve the scalability of map matching techniques used by various robot vision tasks. We propose a first method explicitly aimed at fast succinct map matching, which consists only of map-matching subtasks. These tasks include offline map matching attempts to find a compact part-based scene model that effectively explains each map using fewer larger parts. The tasks also include an online map matching attempt to efficiently find correspondence between the part-based maps. Our part-based scene modeling approach is unsupervised and uses common pattern discovery (CPD) between the input and known reference maps. This enables a robot to learn a compact map model without human intervention. We also present a practical implementation that uses the state-of-the-art CPD technique of randomized visual phrases (RVP) with a compact bounding box (BB) based part descriptor, which consists of keypoint and descriptor BBs. The results of our challenging map-matching experiments, which use a publicly available radish dataset, show that the proposed approach achieves successful map matching with significant speedup and a compact description of map data that is tens of times more compact. Although this paper focuses on the standard 2D point-set map and the BB-based part representation, we believe our approach is sufficiently general to be applicable to a broad range of map formats, such as the 3D point cloud map, as well as to general bounding volumes and other compact part representations.**

I. Introduction

Map matching is a technique in which a robot vision system creates a map of its local surroundings and compares that local map to N global maps previously constructed from prior sensor data. This technique is the foundation for a wide spectrum of robot vision applications, including viewpoint localization, change detection, alignment, merging, map segmentation, and multi-robot mapping [1]–[3]. These applications are built on a common pipeline consisting of stages for extracting features from maps (e.g., SIFT, shape context, and GIST), performing visual indexing and map database searches to find correspondences (e.g., kd-trees, LSH), and map matching to find correspondence sets (points, lines, primitives, and so on) that are inliers to an affine, homography, or geometric transformation, such as by RANSAC matching [4]–[6]. Another key challenge relates to the recent explosion in global efforts to develop scalable map-building systems. To deal with map matching of such large-scale maps, it is necessary to exploit the compact description of data [3]. To this end, most of previous efforts have focused on compact description of feature data (that is, the first and second stages) relying on PCA or other advanced dimension-reduction techniques. Thus far, few studies have focused on the third stage, the map-matching stage using the compact description of *map data*, which we address in this paper (Fig. 1).

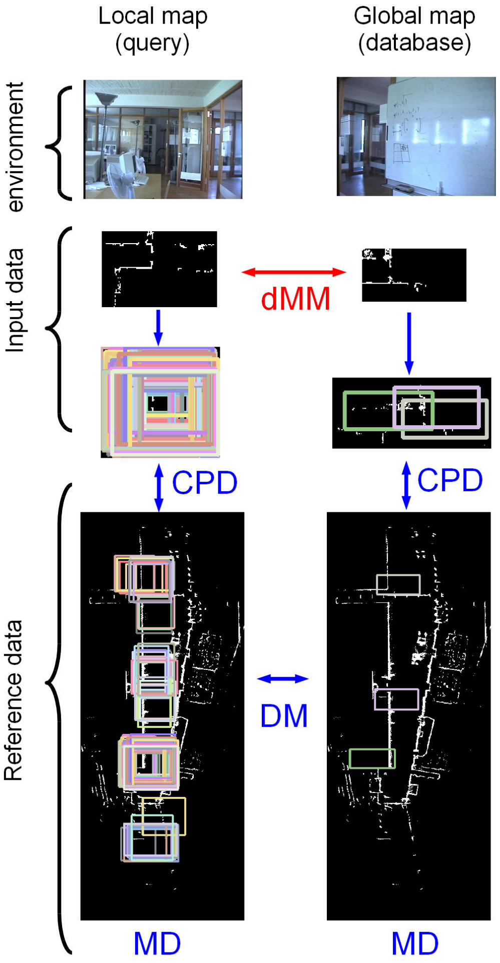

Fig. 1. Compared to existing direct map matching ("dMM") methods, our method indirectly matches between the local map and each of the global maps, while using a known reference map as the intermediary. In offline work, a common pattern discovery ("CPD") process translates an input local/global map to a compact part-based map descriptor ("MD"), by extracting representative parts (colored bounding boxes) that effectively explain an input map from a known reference map. In online work, a descriptor matcher ("DM") rapidly matches between the compact part-based maps.

This work was partially supported by MECSST Grant (23700229, 30325899), by KURATA grants and by TATEISI Science And Technology Foundation.

S.Hanada and K.Tanaka are with Graduate School of Engineering, University of Fukui, Japan. `tnkknj@u-fukui.ac.jp`

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

Our research in this paper is motivated by two independent techniques derived from the field of computer vision: part model [7]–[9] and common pattern discovery [10]–[12]. First, our basic observation is that a part model is a powerful discriminative model, in which a scene is explained by its parts and their configuration, and a part model can also be *compact* if a scene is explained by *fewer larger parts*. Second, we are inspired by the recent progress in common pattern discovery (CPD), which is the mining of common patterns among different scenes. Our key approach is to use the CPD techniques as a method for mining *fewer larger parts* that effectively explain an input scene from a known reference map. In contrast to existing supervised part models that rely on pre-trained part detectors, our CPD-based approach learns a compact part model with *fewer larger parts* in an unsupervised manner, which enables a robot to learn a compact map model without human intervention.

To develop the above map-matching framework, we must address the following questions.

1) How to efficiently mine the parts?
2) How to compactly describe each part?
3) How to efficiently match between part-based maps?

To efficiently mine the parts, randomized visual phrase (RVP) [11], which is a recently developed CPD technique, is adapted for the map data. From our unsupervised modeling standpoint, RVP has advantages over existing CPD techniques in that RVP does not rely on a good segmentation technique, and it does not require a priori knowledge of how many common patterns exist in the scene. To compactly describe each part, we exploit traditional bounding box (BB) based object annotation and knowledge transfer and compactly represent a part as a BB. To efficiently match part-based maps, we can make online matching fast and succinct (descriptor matcher (DM) in Fig.1) because the map matching becomes a low dimensional problem of matching between bounding boxes, and because the CPD can be done in offline as a map-building task. The results of our challenging map-matching experiments, which uses a publicly available radish dataset [13], show that the proposed approach achieves successful map matching with significant speedup and a compact description of map data that is tens of times more compact.

A. Overview of proposed method

The major undertaking of this work can be summarized as follows. (a) We propose a first method explicitly aimed at *fast succinct map matching*, which purely consists of map-matching subtasks. These tasks include offline map matching ("CPD"), which attempts to explain an input map by fewer larger parts, and online map matching ("DM"), which efficiently finds correspondence between the part-based maps. (b) Our part-based scene modeling is *unsupervised* and uses CPD between the input and a known reference maps, which enables a robot to learn a compact map model without human intervention. (c) We present a practical implementation that exploits randomized visual phrase (RVP), a state-of-the-art CPD technique, and employs compact bounding box (BB) -based object annotation. Although this paper focuses on the standard 2D point-set map and the BB-based part representation, we believe our approach is sufficiently general to be applicable to a broad range of map formats, such as the 3D point cloud map, as well as to general bounding volumes and other compact part representations.

B. Relation to Other Work

In the field of computer vision, various types of part models, part detectors, and part appearance models with rigid/deformable templates have been studied with and without learning [7], [8], [14]. In [7], a computational part-based model is introduced as a descriptive model of pictorial structures in an object matching task. In this model, templates of object's parts (such as hair, eyes, mouth for face model) and their configuration are used to determine whether or not the target object is present in the scene. In the above context, template matching techniques have been studied in vision applications using a part-based model, such as object detection/recognition, object comparison, object scaling, and so on. In [8], a framework is presented for joint categorization and segmentation of object images. In this framework, a number of interest points are extracted and compared against the codebook of patches of local appearances: probabilistic voting and refinement are then done to segment the object from the background. In [14], an object detection system based on a mixture of multiscale deformable part models is presented. This system, which relies on discriminative training of classifiers, achieves state-of-the-art results in object recognition tasks. In [15], a framework is presented for weakly supervised discovery of common visual structure in highly variable cluttered images by introducing deformable part-based models. In [16], a method is presented for specifying which region model is assigned to each part by introducing a latent variable. However, existing part-based methods primarily deal with a supervised setting in which a scene category label is explicitly given for training the part detectors. Those supervised approaches are not well suited for our autonomous robot scenarios in which no category or class label is available as training data; rather, a part model with fewer larger parts must be learned from raw map data in an unsupervised manner.

Principal component analysis (PCA) and other advanced dimension reduction techniques have been widely used for the compact description of feature data for the feature extraction and the visual search stages to accelerate map-matching systems. In [6], for example, a method for appearance-guided monocular structure-from-motion for initial motion estimation is presented. In this method, a place recognition scheme and loop closing are employed for loop detection, which works with a visual word-based approach. Earlier works have also presented dimension reduction techniques, specifically locality sensitive hashing [17] and compact projection [18], within RANSAC map matching for large-scale applications [19]. For map matching stage, however, the compact description of feature data is of minor importance, and improving visual search performance is not our objective.

The mining of common patterns among scenes, or CPD [10], remains a challenging task because of huge search scales and problem domains. Several different solutions that address the common problem of rotation, translation, occlusion and segmentation have been studied by employing various techniques, including the earth mover's distance, co-segmentation, and correspondence growing. In [12], a robust CPD technique was also developed based on the correspondence growing framework in the form of a probabilistic Markov Chain Monte Carlo (MCMC) algorithm. However, most of existing frameworks focus on segmentation and discovery of repetitive objects or object parts [20]. In contrast, the current study focuses on *use* of the CPD technique as an unsupervised method for compact part-based scene modeling (see Fig.1).

II. Problem

The map-matching problem addressed in this study is a general 1-to-N matching problem. In it, a local map is given as a query, and the system retrieves a size N global map database to find the relevant global map in terms of the similarity score, under affine, homography, or geometric transformation. In our experiments, we focus on the simple scenario of map matching of 2D point-set maps under rigid transformation. In this scenario, each point $x \in X$ in a map X is represented by a 2D coordinate. The similarity score v between a pair of pointset maps (X,Y) is obtained by searching for the optimal similarity transform parameters $T = (t,o)$ where (t,o) denote the transformation (i.e. translation and orientation) parameters:

$$v = \max_{T} \sum_{x \in X} P_Y(T(x)); \quad P_Y(x) = \begin{cases} 1 & x \text{ is an inlier of } Y \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

A. Performance Metrics

In this study, we develop several different implementations of the 1-to-N map-matching system, and we compare and evaluate them using a standard performance measure of averaged normalized rank (ANR) [21]. To determine ANR, a number of independent map-matching tasks with different queries and databases are conducted. For each task, the rank assigned to the ground-truth global map by a map matcher of interest is investigated and normalized by the database size N. ANR is then obtained as the average of the normalized ranks over all the map-matching tasks.

III. Approach

A. Map Matching Schemes

Three types of map-matching schemes are considered in this paper (Fig.2). The first one is the well-studied *direct map-matching (dMM)* scheme, which takes a pair of maps as input and aims to find correspondences that are inliers to a geometric transformation within a hypothesize-and-test framework. We implement a standard map-matching method based on a RANSAC matching algorithm and use it as a baseline method. In the 2D map-matching scenario, the dMM method iteratively

1) hypothesizes a rigid transformation with rotation and translation
2) tests the hypothesis by counting inlier points under the hypothesized transformation

for a number of times, and, after the iteration, the hypotheses are ranked in descending order of the inlier count.

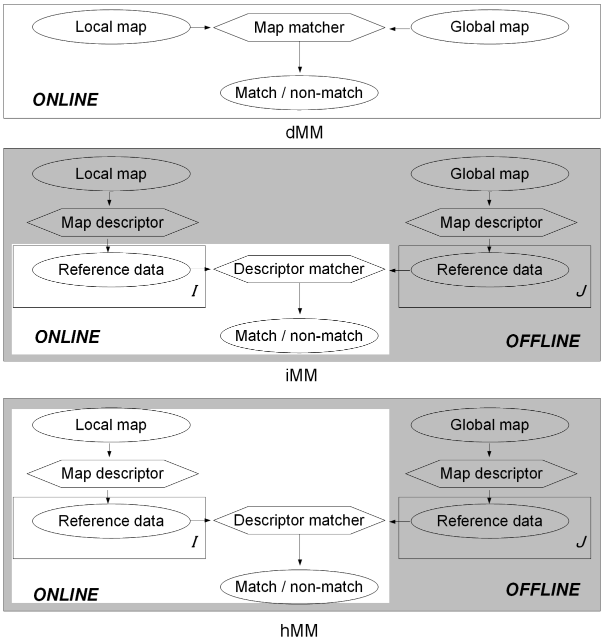

Fig. 2. Three types of map matching schemes considered in this study: direct map matching (dMM), indirect map matching (iMM) and hybrid map matching (hMM). The dMM scheme directly matches between a given map pair, while the iMM and the hMM schemes match in an indirect manner using a given dictionary map as intermediate. While iMM deals with a situation where only the compact map descriptors are available, hMM addresses a situation where the original local map data is available.

The second map-matching scheme we consider is *indirect map matching (iMM)*. This scheme is based on a compact description of map data, and it matches a given map pair in an indirect manner using the dictionary map as intermediatry. The iMM scheme consists of three steps:

1) the common pattern discovery (CPD) mines similar common patterns (i.e., parts) between each input local/global map and the dictionary map;
2) the map descriptor (MD) selects the best parts that effectively explain an input query/database map;
3) the descriptor matcher (DM) evaluates the likelihood that a given query map and a database map are a match pair;

then it ranks the global maps in terms of the likelihood score. Fig.1 illustrates the relationship among CPD, MD and DM.

The third one is *hybrid map matching (hMM)*, which is a hybrid of the above direct and indirect map matching. This scheme is motivated by an asymmetric relationship between the query and the database maps. For example, in a situation where the map matching is conducted online while building the local map, the original local map can be used as a query map, while the original database maps are not available because of memory limitations. The algorithm of hMM is basically the same as that of iMM. A key difference between iMM and hMM is the number of parts that are allowed per map (Fig.2). In the latter case, we are allowed to extract a large number (e.g., 100) of parts from the original local map that is available online.

The following subsections explain the building blocks of the proposed map-matching framework: common pattern discovery, map descriptor, and descriptor matcher. Several different combinations of strategies will be evaluated in the experimental section.

B. Common Pattern Discovery (CPD)

Inspired by the randomized visual phrase (RVP) [11], a recently developed CPD technique, we randomly partition an input map into overlapping bounding boxes (BBs) and view each BB as a potential candidate for the parts. The CPD task is therefore formulated as the problem of finding the *best* BB pairs to crop parts. In this problem,

- the *keypoint BB* crops the input query/database map to indicate the location of a part to be explained out; and
- the *descriptor BB* crops the dictionary map to indicate the appearance of the part.

We believe that good parts should meet the following requirements:

- Maximality Criteria (MC),
- Appearance Similarity (AS), and
- Geometric Criteria (GC).

In MC, each part should explain a large portion of the input map. In AS, the appearance of each part should be *similar* between the input and the dictionary maps. And in GC, the constellation of parts should be consistent between the maps.

To address the above requirements, we first retrieve BBs that satisfy MC and GC; among the retrieved BBs, we select the one that receives the highest AS. The MC criteria judges whether or not the ratio of points within the cropped part to all input map points exceeds a pre-defined threshold $T^{size} = 0.9$. The GC criteria judges whether or not each pair of descriptor BBs overlaps. For each candidate part pair that satisfies both MC and GC criteria, we construct a temporal occupancy grid map R from the dictionary map with a fixed resolution (in implementation, 0.1 m), and we evaluate the AS between the descriptor BB pair $(B, \hat{B})$:

$$f^{AS}(B,\hat{B}) = \max_{T} \sum_{x \in X} P_R(T(x)), \tag{2}$$

where the meanings of P_R and $T(\cdot)$ are the same as in (1).

C. Map Descriptor (MD)

The map descriptor obtains a compact description of an input query/database map. For example, given a given number of parts output by the CPD (in implementation, 100 parts, as shown in Fig.4), each part is represented by a pair of BBs (i.e., a keypoint and a descriptor BBs) as mentioned earlier. Among them, the map descriptor selects a small number of K parts that effectively and compactly explain the input map. Currently, we use a simple selection scheme in which the given parts are ranked in descending order of the AS score; the K top-ranked parts are then output. In our experiments, we test several different settings of K and investigate their influence on the map-matching performance.

Our approach enables the user to make a trade-off between accuracy and compactness by adjusting the number of parts K per input map. A part is compactly represented by the pose and the shape of a BB. Currently, each x, y coordinate of the pose is represented in spatial resolution 0.1 m and memory space consumed per part is 42 bit. For instance, when an input map is represented by 3 parts, the space required per input map is 3×42 bit=126 bit $<$ 16 byte, which is an extremely compact map descriptor.

D. Descriptor Matcher (DM)

We now describe how to efficiently evaluate similarity between map descriptors. Recall that a local map and global maps are described by collections of descriptor BBs, $\{B_l^i\}$ and $\{B_g^j\}$. Intuitively, a pair of descriptor BBs with larger overlap indicates that it is likely that the regions on the global and local maps cropped by the BB pair are similar to each other, and vice versa. To evaluate similarity between a pair of such local and global maps, we introduce a measure of pair-wise similarity between a BB pair

$$f^{RS}(B_l^i, B_g^j) = \frac{1}{\left(Size(B_l^i)Size(B_g^j)\right)^{1/2}} Size(Overlap(B_l^i, B_g^j)), \tag{3}$$

where $Size(\cdot)$ is the area of a given BB on the 2D map, and $Overlap(\cdot,\cdot)$ is the overlap between a given BB pair. Based on the terminology, we design three types of evaluation measures,

Max-max:

$$F^{MM} = \max_i \max_j f^{RS}(B_l^i, B_g^j) \tag{4}$$

Sum-max:

$$F^{SM} = \sum_i \max_j f^{RS}(B_l^i, B_g^j) \tag{5}$$

Sum-max-weighted:

$$F^{SMW} = \sum_i \max_j f^{RS}(B_l^i, B_g^j) \left(f^{AS}(B_l^i, \hat{B}_l^i) \right) \tag{6}$$

The intuition behind the third measure in (6) is that we weight the pair-wise similarity between elements in a BB pair according to the approximation accuracy, which is measured by reusing the appearance similarity f^{AS} described in (2).

IV. Experiments

We conduct challenging map-matching experiments to validate the benefits of the proposed approach. In the following, we first describe the datasets and the map-matching tasks used in the experiments, and we then present our results on performance comparison among different map-matching schemes and strategies, time and space efficiency.

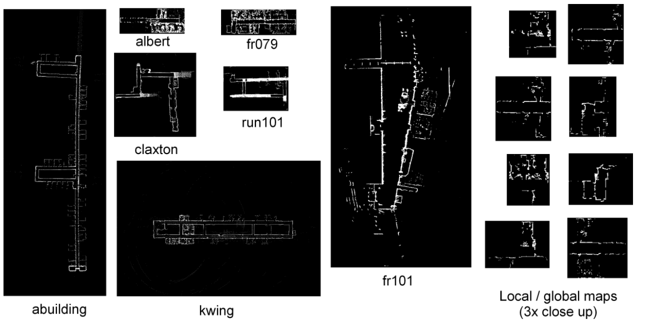

Fig. 3. Datasets used for experiments. “abuilding”, “albert”, “claxton”, “fr079”, “run101”, and “kwing” from radish dataset [13] are used as the dictionary maps. “Local/global maps” are several samples from local and global maps.

A. Dataset

For map matching, we have created a large-scale map collection from the publicly available radish dataset [13], which consists of logs of odometry and laser data acquired by a car-like mobile robot in indoor environments. We created a collection of local/global maps using a scan matching algorithm from each of 5 different datasets—named fr101, albert, kwing, fr079 and abuilding—which respectively were obtained by the mobile robot’s 79–295m travel corresponding to 521-5299 scans. We also used fr079, run, claxton2, albert and fr101, respectively, as the dictionary maps for each dataset.

Fig.3 shows the original datasets and random samples from local and global maps. The map collection consists in total of more than 13,000 submaps. Using the entropy minimization technique in [22], the *xy*-axes of each map were aligned with the “Manhattan-like structure”. As shown in Fig.3, our map collection contains many near duplicate maps, which makes the map matching a challenging task.

B. Map Matching Tasks

Recall that the objective of map matching is giving a local map as a query to find a relevant map from the global map database. The relevant map is defined as a global map that satisfies two conditions: 1) Its pose is near the query map’s pose within a pre-defined range, where the pose of a map is defined as the center of gravity of the map’s point cloud, and 2) Its distance traveled along the robot’s trajectory is distant from that of the local map, such as in a “loop-closing” situation in which a robot, after traversing a loop-like trajectory, returns to a previously explored location.

For each relevant map pair, a map-matching task is conducted by using a local map and a size N global map database, which consists of one relevant map and $(N-1)$ random irrelevant maps. The spatial resolution of the temporal occupancy map used by the iMM and hMM is set to 0.1 m. The descriptor matcher uses “Sum-Max” and “Sum-Max-Weighted” as the default strategies, respectively, for iMM and hMM. We have implemented several combinations of map-matching algorithms in C++, and successfully tested on various maps.

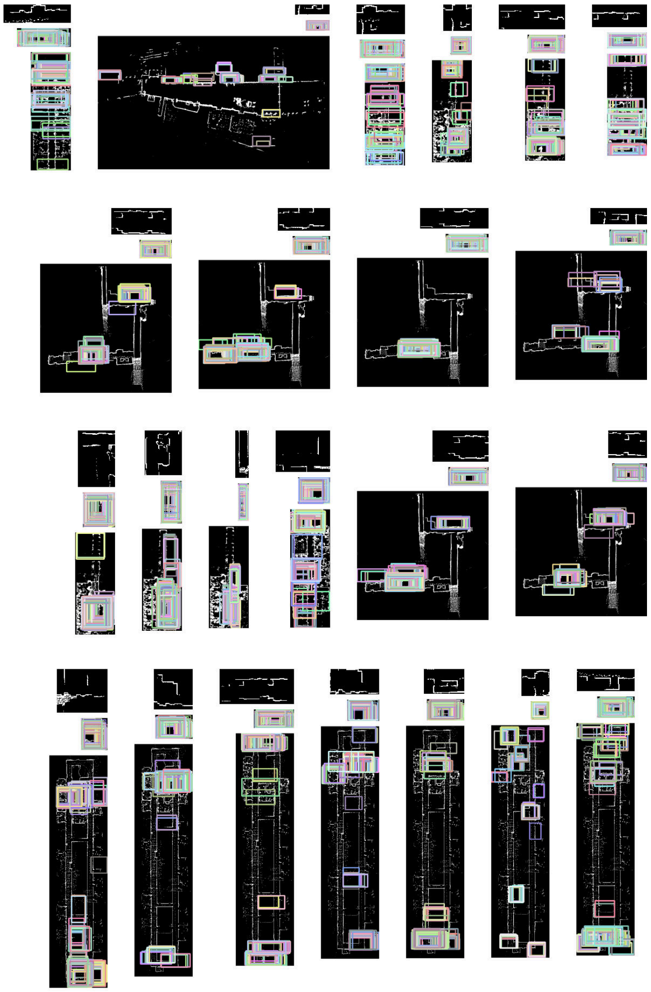

Fig. 4. Results of part-based scene modeling. For each panel, the top subpanel shows the input local map, while the middle and the bottom subpanels respectively illustrate the representative parts discovered w.r.t. the local and the global maps. In each figure, the white dots indicate point cloud in the map, and the colored rectangles indicate bounding boxes that crop the representative parts.

Shown in Fig.4 are the results of part-based scene modeling, in which an input local/global map is explained by a pool of representative parts discovered by CPD. The CPD task was mostly successful, and similar input maps were characterized by similar descriptor BBs. Quantitative evaluations of our CPD-based approach will also be provided in the following subsections.

C. Comparison among dMM, iMM, and hMM

For performance comparison, we evaluated the averaged normalized rank (ANR) introduced in Section II-A for all the three basic map-matching methods: dMM, iMM and hMM. All map-matching tasks were conducted using 13,592 different local and global map databases. Fig.5 shows the normalized cumulative histograms of rank assigned to the relevant map, and Table I summarizes the ANR performance.

The proposed hMM achieved results comparable result to the dMM method when the number of parts per map K was set to 3–5, despite the fact that the proposed hMM achieves

TABLE I

SUMMARY OF ANR PERFORMANCE.

	Data ID				
Method	#1	#2	#3	#4	#5
Max-Max hMM cpd:3	36.2	30.9	40.1	32.8	34.1
Sum-Max hMM cpd:3	35.3	27.6	36.5	30.2	33.4
rerank10 hMM cpd:3	34.0	26.8	36.4	27.5	32.5
rerank20 hMM cpd:3	33.4	26.5	**34.3**	27.4	32.0
dMM	32.9	26.1	**34.3**	**27.1**	31.6
iMM cpd:1	44.5	51.0	53.0	55.8	42.1
iMM cpd:2	38.4	36.1	41.1	38.6	37.0
iMM cpd:3	35.3	28.5	39.1	31.9	33.3
iMM cpd:4	33.8	25.2	36.9	30.4	30.3
iMM cpd:5	**32.0**	**24.1**	35.9	29.5	**29.5**
hMM cpd:1	33.3	30.2	40.6	30.5	30.2
hMM cpd:2	33.6	26.8	36.4	27.4	31.9
hMM cpd:3	**32.0**	26.3	35.7	28.1	30.0
hMM cpd:4	33.7	26.6	36.5	27.4	32.4
hMM cpd:5	32.8	26.0	37.0	27.4	31.4

The Data ID #1-#5 correspond to ANR for global map database from "fr101", "abuilding", "albert", "kwing" and "fr079", where "fr079", "fr101", "run", "claxton2" and "albert" are respectively used as dictionary map. The colored fonts and the bold-face fonts respectively represent the best 5 methods and the best method for each data.

tens of times more compact map representation than dMM, as will be shown in Section IV-G. Much of the analysis from the previous sections on map matching also holds for the dataset considered here. By extracting a number of parts from the query local map and comparing their attributes (i.e., BB pairs) against each map in the global map database, our method is able to successfully perform the map matching. On the other hand, the presented iMM method is satisfactory when we set the number of parts per input map to a large value, 5. This is because the input map tends to be well explained by the larger set of common object patterns, which is still a compact representation, as will be discussed in Section IV-G.

D. Effect of Matching Strategies

We also compared three different implementations of the descriptor matcher for the hMM method, which was explained in Section III-D. However, the difference among them is turned out to be subtle, as shown by "Max-Max hMM", "Sum-Max hMM", and "hMM" in Fig. 5 and in Table I.

E. Effect of Reranking

For precise map matching, we tested another "reranking" strategy, which is in essentially a cascade of dMM and hMM. The strategy is motivated by the observation that, according to Fig. 5, the standard dMM method is good at ranking top-ranked elements more precisely than the hMM method. Using the cascaded method, the top R elements ranked by the proposed hMM method were further input to the dMM method, and their rankings were reordered according to the match score assigned by the dMM method. We set the number R of elements to be reranked as $R = 10$ and $R = 20$ and conducted the map-matching tasks. The result is similar or better than the proposed hMM method, as shown in Fig.5. On the other hand, a major drawback of this strategy is memory usage, as the dMM method that the strategy employs requires the information from the original global maps.

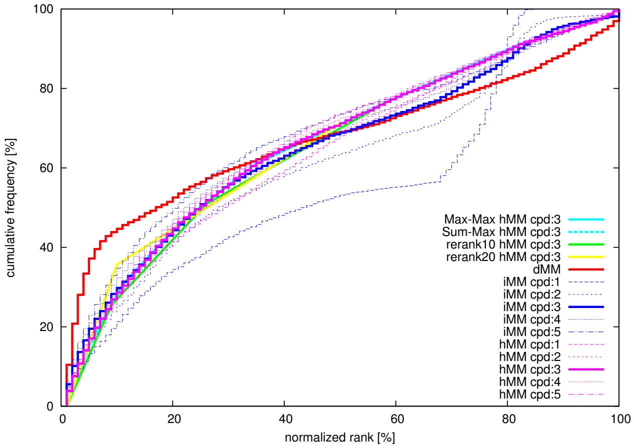

Fig. 5. Normalized rank for each strategy.

F. Time Efficiency

Matching between a query of a local map and a database map took less than 5 ms on a 2.4 Ghz Intel laptop. The time complexity was linear in terms of the number of parts per map and the database size. We tested different settings of the number of parts per map, 1, 2, 3, 4 and 5, and got time cost 1.3, 2.3, 3.3, 4.2 and 5.0 msec. In contrast, the conventional dMM algorithm took an average of 33 msec. Its complexity was linear in the database size, and its time cost could be quite high when using a large size database, e.g., with tens of millions of landmarks. On the other hand, one of main concerns of our method is the time cost for CPD. The cost in principle depends on the size of the dictionary map, this can be non-negligible when using a large-scale dictionary map. However, because cost is not dependent on database size, it therefore does not affect the scalability of our proposed approach. It is beyond the scope of this paper to discuss how to suppress the cost for CPD performed as part of the map-building task, and it will have to be addressed in future work.

G. Space Efficiency

Space cost for our map descriptor is linear to the number of parts per map and the cost per part. In general, a part is defined by a pair of BBs, i.e., a keypoint BB $[x_{begin}, x_{end}] \times [y_{begin}, y_{end}]$, and a descriptor BB $[\hat{x}_{begin}, \hat{x}_{end}] \times [\hat{y}_{begin}, \hat{y}_{end}]$. Because the shapes of both BBs are the same, our database omits $(\hat{x}_{end}, \hat{y}_{end})$ as redundant information. In implementation, 7 bits are used for each of x_{begin}, x_{end}, y_{begin}, y_{end}. (x_{begin}, y_{begin}) indicates a point on the dictionary map and is represented in 14 bits. In total, $7 \times 4 + 14$ = 42 bits are used for each part. For instance, the proposed iMM method consumes 42×3 bits = 126 bits when it uses 3 parts per map. Compared with the dMM method, which costs $(7+7) \times 500$ = 7000 bits for typical size 500 point cloud map, our map-matching method achieves compact description of map data that is tens of times more compact.

V. Conclusion and Future Work

In this paper, we proposed a first method explicitly aimed at *fast succinct map matching*, which consisted only of map-matching subtasks. These tasks included offline map matching (CPD) attempts to explain an input map using *fewer larger* parts, and online map matching (DM) to efficiently find correspondence between the part-based maps. Our part-based scene modeling was unsupervised and used CPD between the input and a known reference maps, which enables a robot to learn a compact map model without human intervention. We presented a practical implementation that leverages the randomized visual phrase (RVP), a state-of-the-art CPD technique, and a compact BB-based part descriptor. We showed that the proposed approach achieves successful map matching with significant speedup and a compact description of map data that is tens of times more compact.

In future work we plan to continue our exploration of CPD, and consider how to achieve good tradeoff between the compactness and the accuracy. We will also investigate efficient methods for building the part-based maps. It would be interesting to extend the methods to support a compressive SLAM task [18], which aims at the incremental building of the compact map model.

Another obvious improvement is to optimize the reference maps. Because our approach is designed to represent an input map by cropped reference maps, it would not be suitable for general cases where whole regions of the input map are dissimilar from the reference map. In the future we shall study a way for automatically choosing the reference maps adaptively for given global maps.

Finally, our approach could be extended to a broad range of map formats, such as the 3D point cloud map, as well as to other compact part representations, such as general bounding volumes. In this direction, the approach was recently extended to an alternative map format, view sequence map, by introducing an unsupervised part-based scene modeling technique, "bag-of-bounding-boxes (BoBB)" [23].

References

[1] Brian Yamauchi and Randall Beer. Spatial learning for navigation in dynamic environments. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 26(3):496–505, 1996.

[2] Shoudong Huang, Zhan Wang, and Gamini Dissanayake. Sparse local submap joining filter for building large-scale maps. *IEEE Trans. Robotics (TRO)*, 24(5):1121–1130, 2008.

[3] Andreas Wendel, Michael Maurer, Gottfried Graber, Thomas Pock, and Horst Bischof. Dense reconstruction on-the-fly. In *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1450–1457, 2012.

[4] Stephen Se, David G. Lowe, and James J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *I. J. Robotic Res.*, 21(8):735–760, 2002.

[5] Mark Cummins and Paul Newman. Highly scalable appearance-only slam - fab-map 2.0. In *Robotics: Science and Systems*, 2009.

[6] Davide Scaramuzza, Friedrich Fraundorfer, and Marc Pollefeys. Closing the loop in appearance-guided omnidirectional visual odometry by using vocabulary trees. *Robot. Auton. Syst.*, 58(6):820–827, 2010.

[7] M.A. Fischler and R.A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Computers*, C-22(1):67 – 92, 1973.

[8] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *Euro. Conf. Compuer Vision (ECCV) workshop on statistical learning in computer vision*, pages 17–32, 2004.

[9] Pablo Arbelaez, Bharath Hariharan, Chunhui Gu, Saurabh Gupta, Lubomir D. Bourdev, and Jitendra Malik. Semantic segmentation using regions and parts. In *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3378–3385, 2012.

[10] Hung-Khoon Tan and Chong-Wah Ngo. Common pattern discovery using earth mover s distance and local flow maximization. In *IEEE Int. Conf. Computer Vision (ICCV)*, pages 1222–1229, 2005.

[11] Yuning Jiang, Jingjing Meng, and Junsong Yuan. Randomized visual phrases for object search. In *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3100–3107, 2012.

[12] Yuuto Chokushi, Kanji Tanaka, and Masatoshi Ando. Common landmark discovery in urban scenes. *IAPR Int. Conf. Machine Vision Applications*, 2013.

[13] Andrew Howard and Nicholas Roy. The robotics data set repository (radish), 2003.

[14] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645, 2010.

[15] Megha Pandey and Svetlana Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, pages 1307–1314, 2011.

[16] S.N. Parizi, J.G. Oberlin, and P.F. Felzenszwalb. Reconfigurable models for scene recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2775–2782, 2012.

[17] Kenichi Saeki, Kanji Tanaka, and Takeshi Ueda. Lsh-ransac: An incremental scheme for scalable localization. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 3523–3530, 2009. `http://rc.his.u-fukui.ac.jp/LR.pdf`.

[18] Tomomi Nagasaka and Kanji Tanaka. An incremental scheme for dictionary-based compressive slam. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 872–879, 2011. `http://rc.his.u-fukui.ac.jp/ICDCS.pdf`.

[19] Kanji Tanaka and Tomomi Nagasaka. Dictionary-based compressive slam. *SICE JCMSI Journal of SICE*, 6(1):54–64, 2013.

[20] Tinne Tuytelaars, Christoph H. Lampert, Matthew B. Blaschko, and Wray L. Buntine. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88(2):284–302, 2010.

[21] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE Int. Conf. Computer Vision (ICCV)*, pages 1470–1477, 2003.

[22] Sven Olufs and Markus Vincze. Robust single view room structure segmentation in manhattan-like environments from stereo vision. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 5315–5322, 2011.

[23] Masatoshi Ando, Kanji Tanaka, and Yousuke Inagaki. A bag-of-bounding-boxes approach to object-level view image retrieval. In *Proc. SICE Annual Conference*, 2013. `http://rc.his.u-fukui.ac.jp/BOBB.pdf`.