Discriminative Map Matching Using View Dependent Map Descriptor

Liu Enfu  
Tanaka Kanji

Abstract—The problem of matching a local occupancy grid map built by a mobile robot to previously built maps is crucial for autonomous navigation in both indoor and outdoor environments. In this paper, the map matching problem is addressed from a novel perspective, which is different from the classic bag-of-words (BoW) paradigm. Unlike previous BoW approaches that trade discriminativity for viewpoint invariance, we develop a local map descriptor that is viewpoint-dependent and highly discriminative. Our method consists of three distinct steps: (1) First, an informative local map of the robot’s local surroundings is built. (2) Next, a unique viewpoint is planned in accordance with the given local map. (3) Finally, a synthetic view is described at the designated viewpoint. Because the success of our local map descriptor (LMD) depends on the assumption that the viewpoint is unique given a local map, we also address the issue of viewpoint planning and present a solution that provides similar views for similar local maps. Consequently, we also propose a practical map-matching framework that combines the advantages of the fast succinct bag-of-words technique and the highly discriminative LMD descriptor. The results of experiments conducted verify the efficacy of our proposed approach.

I. INTRODUCTION

The problem of matching a local occupancy grid map built by a mobile robot to previously built maps is crucial for autonomous navigation in both indoor and outdoor environments [1]–[7]. This paper addresses a general 1-to-N matching problem in which a 2D pointset map is given as a query, and the system searches over a size \( N \) map database to find similar database maps that are relevant under rigid transformation.

The classical approach to the map-matching problem is to describe the appearance of each local map using high-dimensional local invariant feature descriptors such as shape features (e.g., polestar feature [8]), and perform feature matching between query and database maps. One major limitation of such an approach is the time consumed comparing the high-dimensional descriptors [9]. One of the most popular approaches used to address this computational cost is the bag-of-words (BoW) approach, in which an unordered collection of vector quantized feature descriptors (e.g., shape context, polestar), which are extracted at random, dense, or interest points (e.g., FLIR), is used for compact map representation and efficient matching to pre-built maps. Thus far, the BoW approach has been utilized in various map-matching tasks, ranging from view image sequence maps to 3D point cloud maps [5]–[7]. Our proposed approach is also built on the BoW system in [10], in which the BoW framework is successfully applied to the retrieval of 2D occupancy maps using rotation invariant polestar descriptors.

In this paper, we consider the local map descriptor (LMD), which involves the generation of text descriptions of local map content to facilitate fast succinct text-based map matching. Unlike previous local feature approaches that trade discriminativity for viewpoint invariance, we develop a holistic view descriptor that is viewpoint-dependent and highly discriminative. Our method consists of three distinct steps:

1) First, an informative local map of the robot’s local surroundings is built.
2) Next, a unique viewpoint is planned in accordance with the given local map.
3) Finally, a synthetic view is described at the designated viewpoint.

The success of our holistic view descriptor is based on the assumption that the viewpoint is unique given a local map. Therefore, we also address the issue of viewpoint planning and present a solution that provides similar views for similar local maps. We also propose a practical map-matching framework that combines the advantages of the fast succinct BoW techniques (e.g., [11]), and the highly discriminative LMD holistic view descriptor. The results of experiments conducted using the publicly available radish dataset [12] and our own collected dataset confirm the efficacy of our proposed approach.

In this paper, we focus on methods that describe not only local feature descriptors but also the local keypoint configuration among them. Among these methods, the part model [13], in which a scene is modeled as a collection of visual parts, is very popular. The model uses information on relative positions as spatial cues to improve the discriminative power of representation. However, existing part-based models primarily focus on a small set of pre-learned parts. Our approach is somewhat similar in concept to the spatial pyramid matching approach in [14], as opposed to the focus on kernel definition and improvement to discriminative power of previous solutions. Most of the works cited above either explicitly or implicitly assume that the viewpoint trajectory of the mapper robot w.r.t. the local map is unavailable. In contrast, we explicitly use the viewpoint information produced by our viewpoint planner as a cue to compute the holistic view descriptor. The success of our approach is based on the assumption that the viewpoint planner provides a unique viewpoint given a local map; therefore, we also consider the issue of viewpoint planning. To the best of our knowledge, these two issues have not been explored in previous approaches.

Our work has been supported in part by JSPS KAKENHI Grant-in-Aid for Young Scientists (B) 23700229, and for Scientific Research (C) 26330297.

E. Liu and K. Tanaka are with Graduate School of Engineering, University of Fukui, Japan, tnkknj@u-fukui.ac.jp
II. LOCAL MAP DESCRIPTOR

A. Baseline System

This section describes the baseline map-matching system, on which our proposed approach is built, and which is also used as a benchmark for performance comparison in the experimental section, Section III. The main steps in the procedure carried out by the system are as follows: (1) Extraction of appearance features from each local map, (2) translation of the extracted features to a BoW descriptor, and (3) construction/retrieval of the map database from the BoW descriptors. These three steps are explained in detail below.

1) Feature Extraction: We adopt the polestar feature for our purpose because it has several desirable properties, including viewpoint invariance and rotation independence, and has proven effective as a landmark for map matching in previous studies [10]. The extraction algorithm consists of three steps (Fig.1): (1) First, a set of keypoints are sampled from the raw 2D scan points. (2) Next, a circular grid is imposed and centered at each keypoint with different radius. (3) Finally, the points falling into each circular grid cell are counted and the resulting D-dim vector outputted as the polestar descriptor.

2) BoW Descriptor: Next, we quantize each D-dim polestar vector to a 1-dimensional code termed “visual word”. This quantization process consists of three steps: (1) normalization of the D-dim vector by the vector’s L1 norm, (2) binarization of each i-th element of the normalized vector into $b_i \in \{0, 1\}$, and (3) translation of the binarized D-dim vector into a code or a visual word: $w_a = \sum_i 2^{a_i} b_i$. Currently, the threshold for binarization is determined as the mean of all the elements of the vector. In consequence, a map is represented by an unordered collection of visual words $\{w_a \mid w_a \in [1,K]\}$, called BoW. Because we consider D-dim binarized polestar descriptors, the vocabulary size is $K = 2^{10}$.

3) Database Construction/Retrieval: We use the BoW representation for both the database construction and retrieval processes. In the former process, each local map is indexed by the inverted file system, by using each word $w_a$ belonging to the map as an index. In the latter process, all the indexes that have words in common with the query map are accessed and the resulting candidate database maps are ranked based on the frequency or the number of words in common. A frequency histogram of visual words is represented by a $K$-dim vector when we have $K$ words in the vocabulary. Similarity between a pair of BoW frequency histograms is evaluated in terms of the histogram intersection.

B. Proposed Extension

In this section, we outline our proposed extension. As mentioned earlier, we built on the baseline system described in Section II-A, and developed a novel holistic view descriptor. Our method consists of three distinct steps: (1) build a local map, (2) plan a unique viewpoint given the local map, and (3) describe a synthetic view at the planned viewpoint. These three steps as well as the modified map-matching algorithm are detailed in below.

1) Map Building: We first build a local map from a short sequence of perceptual and odometry measurements; each measurement sequence must be sufficiently long to cover rich photometric and geometric information about the robot’s local surroundings. In implementation, each sequence corresponds to the robot’s 3 m run. Any map-building algorithm (e.g., FastSLAM, scan matching) can be used to register a measurement sequence into a local map. We start a local map every time the robot’s viewpoint moves along the path. This results in a collection of overlapping local maps along the path.

2) Viewpoint Planning: We wish to design a robust planner that provides a unique viewpoint given a local map. (Note that the viewpoint is not necessarily one of the actual viewpoints.) An occupancy grid map is constructed from the 2D pointset map and used as input to our viewpoint planner. Currently, we plan the unique viewpoint near to the center of gravity (CoG) of all the occupancy grid cells. This strategy is inspired by the observation that the CoG can be unique given a local map both in narrow corridors and in rooms.

In implementation, all the viewpoints on the free space cells on the local occupancy grid map are viewed as candidate viewpoints, and among them, the closest candidate to the CoG is selected as the viewpoint for the holistic view descriptor. Subsequently, we determine the viewing direction based on the “dominant direction” [15] of the occupancy grid cells. An intuitive example of the dominant direction is...
Manhattan world-like environments, where the two dominant directions should be the two orthogonal directions of the manhattans world. To estimate the dominant directions, we adapt the Manhattan world assumption criteria in [15].

3) Holistic View Descriptor: Let us now look at the holistic view at the planned viewpoint and represent it in the BoW form. A key difference of our BoW representation from that of previous works is that we no longer need to rely on view invariant local features that trade discriminativity for view invariance. Instead, we can exploit the knowledge of viewpoint w.r.t. the ego-centric local map coordinate to make the holistic descriptor view-dependent, and thus highly discriminative. Our BoW representation comprises appearance words and pose words. The former represents the appearance descriptor of each local feature w.r.t. the local map coordinate. Currently, we simply use the descriptor of each local feature and quantize it into an appearance word, as we did in Section II-A.2. The latter, pose word, represents the keypoint of each local feature w.r.t. the local map coordinate. During implementation, we quantize the keypoint \((x, y)\) w.r.t. the local map’s coordinate to obtain the pose word \((w_x, w_y)\) with resolution quantization step size of 0.1 m. As a result, our visual word is in the form:

\[
(w_x, w_y, w_a),
\]

4) Map Matching: To index and retrieve the BoW map descriptors, we use the appearance word \(w_a\) as the primary index for the inverted file system, while using the pose word \((w_x, w_y)\) as an additional cue for fine matching. The retrieval stage begins with a search of the map collection using the given appearance word \(w_a\) as a query to obtain all the memorized feature points with common appearance words, and filter out those feature points whose pose word \((w'_x, w'_y)\) is distant from that of the query feature \((w_x, w_y)\):

\[
|w'_x - w_x| > D_{x,y},
\]

\[
|w'_y - w_y| > D_{x,y},
\]

to obtain the final shortlist of maps. Currently, we use a large threshold, \(D_{x,y} = 1[m]\), to suppress false negatives, i.e., incorrect identification of relevant maps as not being relevant.

III. EXPERIMENTS

We conducted map-matching experiments to verify the efficacy of the proposed approach. In the ensuing subsections, we first describe the datasets and the map-matching tasks used in the experiments, then present the results obtained and conduct performance comparison against the baseline system.

A. Dataset

For map matching, we created a large-scale map collection from the publicly available radish dataset [12], which comprises odometry and laser data logs acquired by a car-like mobile robot in indoor environments (Fig. 2). We created a collection of query/database maps using a scan matching algorithm from each of six different datasets—namely, “abuilding,” “albert,” “fr079,” “run,” “fr101,” and “kwing”—which were obtained by the mobile robot’s 79–295 m travel, corresponding to 521–5299 scans. Fig. 3 shows examples of the query and database maps. The map collection comprises more than 13,000 maps. Our map collections contain many virtually duplicate maps, which makes map matching a challenging task.

B. Qualitative Results

Recall that the objective of map matching is to find a relevant map from the map database for a local map given as a query. The relevant map is defined as a database map that satisfies two conditions: (1) Its pose is near the query map’s pose within a predefined range, where the pose of a map is defined as the CoG of the map’s pointset; and (2) its distance traveled along the robot’s trajectory is distant from that of the query 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 using a query map and a size \(N\) map database, which consists both of the relevant map and \((N - 1)\) random irrelevant maps. The spatial resolution of the occupancy map is set to 0.1 m. We implemented the map-matching algorithm in C++, and successfully tested it on various maps. Figs. 3 show the results of map matching using the baseline (“BoW”) and the proposed (“LMD”) systems. As can be seen, fewer false positives appear in the case of the proposed LMD method than the BoW method. This is because many of the incorrect matches are successfully filtered out by the proposed feature, which uses the keypoint configuration as a cue. Quantitative evaluation results for our approach are provided in the next subsection.

C. Quantitative Results

For performance comparison, we evaluated the averaged normalized rank (ANR) [16] for both the BoW and LMD methods. ANR is a ranking-based performance measure in which a lower value is better. To determine ANR, we conducted a number of independent map-matching tasks with different queries and databases. For each task, the rank assigned to the ground-truth database map by a map matcher of interest was investigated and normalized by the database
size $N$. ANR was subsequently obtained as the average of the normalized ranks over all the map-matching tasks. All map-matching tasks were conducted using 13,592 different queries and map databases.

Table I and Fig.4 summarize the ANR performance. The proposed LMD system clearly outperforms the baseline BoW system. By filtering out incorrect matches using the keypoint configuration as a cue, the LMD method was able to successfully perform map matching in many cases, as shown in the figure. In contrast, the BoW system based on appearance words alone often does not perform well, mainly because of the large number of false matches. The above results verify the efficacy of our approach. Table I also reports results of additional experiments using our own collected dataset, termed “OCD” in the table. In this experiments, we used pioneer 3 DX mobile robot equipped with LMS200 laser scanner and collected a set of 38 local maps in our university building 1F, 2F, and 5F. All the maps used in the experiment are shown in Fig.5 and shown in Table I are ANR performance. As can be seen, the proposed method outperforms the previous BoW method for almost all maps considered in the experiments.

IV. CONCLUSIONS

In this paper, we focused on generating text description of local map content for fast succinct text-based map matching. In particular, we presented a novel holistic view descriptor that describes a synthetic view at a planned viewpoint. We addressed the issues involved in building a local map, planning viewpoints, and computing the holistic view descriptor. The results of experiments conducted with the publicly available radish dataset confirm the efficacy of our proposed approach. In the future, we plan to use the

![Fig. 4. ANR performance for each dataset (horizontal axis: sorted query map ID, vertical axis: ANR in [%]).](image)

**TABLE I**

<table>
<thead>
<tr>
<th>dataset</th>
<th>abuilding</th>
<th>albert</th>
<th>fr079</th>
<th>fr101</th>
<th>kwing1</th>
<th>run1</th>
<th>OCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW</td>
<td>29.3</td>
<td>35.0</td>
<td>24.0</td>
<td>32.6</td>
<td>18.7</td>
<td>41.7</td>
<td>37.1</td>
</tr>
<tr>
<td>LMD</td>
<td>7.0</td>
<td>26.6</td>
<td>17.1</td>
<td>16.7</td>
<td>3.6</td>
<td>15.2</td>
<td>22.6</td>
</tr>
</tbody>
</table>

![Fig. 3. Examples of map matching. Left: relevant map pairs. Right: irrelevant map pairs.](image)
Fig. 5. Results for own collected dataset (OCD). Shown in the figure are from top to bottom, pairs of query map #0-#4, #5-#8, #9-#11, #12-#14, #15-#18, #19-#24, #25-#30, #31-#34, and #35-#38, and its ground-truth database map. Performance is evaluated in terms of rankings of ground-truth database map and the ranking values are shown in the figure for either proposed LMD method (“M2T”) or conventional BoW method (“BoW”). The details are better seen by zooming on a computer screen.

Fig. 6. Ranks for each query from our own collected dataset. (horizontal axis: query map ID, vertical axis: ANR in [%]).

presented LMD system for long-term operation of robots in familiar environments. Although this paper focused on the standard 2D pointset map, 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 general view-based maps.

REFERENCES


