Cognitive Navigation for Indoor Environment Using Floorplan*

Jun Li, Chee Leong Chan, Jian Le Chan, Zhengguo Li, Kong Wah Wan, and Wei Yun Yau

Abstract— **The recent years have seen the increasing importance of cognitive models for improved robot navigation. In this paper, a novel cognitive navigation package, which consists of topometric map representation and a three-level path planner, is proposed. The topometric maps are built from architectural floor plans with additional features within a limited number of selected regions. The inherent discrepancies between floor plans and the robot's actual sensory readings are handled by the three-level path planner. The unique feature of this approach is that accurate localization is only required at the selected regions. At the other regions the robot will rely on the guiding directions towards the goal rather than on its accurate position on the map for navigation. Experiments show that our approach can endow a robot with capability for semantic interpretation and localization in unseen and dynamic environments.**

I. INTRODUCTION

In robot navigation, the representation of the environment plays a critical role for localization and path planning. There are two main categories of the map representation: metric map, and topological map [1], [2].

Metric map: considers a geometric space in which environment information is encoded in precise coordinates. These precise maps are most commonly used by robots. However, it is not easy to build and maintain such a map [3] due to factors in the environment such as feature-less regions, repetitive patterns, dynamic and frequent changes. Furthermore, when the environment is large, it is difficult to build a globally consistent map because of accumulated errors [4].

Topological map: considers a graphic representation of the environment where nodes indicate places and edges indicate the relationship among nodes. The topological representation is concise and compatible with how humans would perceive the environment. However, relying on a topological map may result in ineffective navigation due to missing metric information and difficulty for path planning.

Existing works on topometric map based navigation in [2], [4]–[7] mainly use topological maps to help localization through densely distributed nodes [6], [7], or segmenting the whole space into different parts as multiple reference frames [4]. However, the former loses the advantage of conciseness in topological representation, while the latter requires precise localization not only locally but also on the whole map.

Humans do not perform precise localization everywhere, and they can navigate using abstract data such as floor plans, verbal and symbolic instructions [8]–[10]. These information is of a topological/directional nature rather than metric detail. When traveling indoor using such abstract maps, humans tend to pay more attention at the intersections (e.g. so as to make the correct turn) than along corridors. This implies that metric processing is sparse compared to keeping direction. One other example is driving on the highway: at or near an exit, the driver makes a metric localization decision to exit or not; in between exits, it is more important to maintain the safe and allowable moving directions than knowing the vehicle's exact location.

Recent research on human spatial navigation [11], [12] indicate that environment representations are related to the scale of space. For vista space (observable from a single location), egocentric representation (objects are represented relative to the perceiver) is more suitable than the allocentric representation (objects are represented relative to the environment) for navigation in confined and dynamic environments. that humans navigate with coarse topological localization to guide local motion planning.

In our literature review, we found that despite their widespread availability in modern buildings, little attention to using 2D floor plans for robot indoor navigation. This is mainly due to the discrepancies between 2D floor plans and on-ground reality. Notwithstanding, using 2D floor plans is beneficial: (1) they reveal the topological relationship of places in the environment in a globally consistent manner; (2) they can be treated as a coarse level map where the discrepancies indicate the areas needing map update.

These observations inspire us to propose a new topometric representation of the environment and to design a direction based navigation approach for mobile robots. Our proposed topometric map is based on the floor plan to represent topological relationship of the whole environment. To maintain map conciseness, nodes are only selected at some critical places, and their visual features are used as metric data utilized by robots to be made aware of their global positions. The location of the robot is represented by a hybrid state with one discrete state to indicate the previous node and a continuous state to indicate the current location with respect to the previous node. To fully utilize the advantages of the proposed topometric map, a three-level path planner is introduced to guide the robot from the starting point to the goal one. At the global level, the topological path will be generated by an efficient global path planner. During traversal, a SLAM map is created to estimate a robot's position on the floor plan between two connecting nodes. The local robot frame is represented using sensor data from which the local region exit points are extracted. To ensure the local motion follow the correct path to the next node,

*This work was supported by the National Robotics Pro-gramme (NRP) under the SERC Grant 192 25 00049

Jun Li, Chee Leong Chan, Jian Le Chan, Zhengguo Li, Kong Wah Wan, and Wei Yun Yau are with Institute for Infocomm Research,A*STAR, Singapore 138632. Corresponding author: Dr. Jun Li, jli@i2r.a-star.edu.sg

a nominal path between nodes is built up as the directional guidance to select the proper exit way-points towards the next node. With the proposed topometric representation and the hierarchical path planner, an efficient cognitive navigation solution is available for indoor service robots.

The main contributions of this work are as follow:

- Use of 2D floor plans: A floor plan is used as the initial topometric map since it is globally and topologically consistent. The metric information is only added at nodes based on their local structures, without the need for tedious exploration of the whole environment.
- Concise topometric map representation: Different from the other topometric representation, we carefully select and build nodes to ensure the number of nodes are small while preserving their functionalities, e.g. enable a robot to make a correct turn towards the next node along the path to the goal. To achieve this, visual features are only added at nodes. Such concise map makes map maintenance easier, since updates are only required at a very limited number of small regions.
- Efficient hierarchical path planning: As metric localization is not always available on our topometric map, the traditional 2-level path planning will fail. Therefore we propose a 3 level path planing as follows:
  - global logical path planning on nodes to address efficiency of the logical path planning without the sensor observation;
  - regional path planning between two adjacent nodes to build up a nominal path using the floor plan map. A group of points in the nominal path are selected to guide the navigation direction;
  - local path planning on the sensor data to address collision free movement. The proper regional exit point is selected using the guide points on the nominal path towards the next node. This is the key component in our topometric map based robot navigation.

The rest of this paper is organized as below. Prior works on floor plan and topometric maps are presented in Section II. Details of our proposed navigation package are provided in Section III. Experimental results are given in Section IV to verify the efficiency of our method. Finally, conclusion remarks and future works are discussed in Section V.

II. Prior Works

Our main contributions are related to floor plan and topometric maps. We briefly describe some relevant works.

A. Floor plan based navigation

2D floor plans are usually available for the modern building. They contain the locations of permanent structures (walls, windows, doors and stairs), and semi-permanent structures (fixtures). There were two focuses for the floor plan analysis: for human navigation and for robot navigation.

1) Floor plan analysis for human navigation: Floor plans were mostly analyzed for human-oriented navigation [8], [13]–[17].

Lorenz *et al.* [14] proposed hybrid spatial model for indoor environment. They labeled the nodes and edges qualitatively as well as quantitatively to support wayfinding and therefore to generate human-oriented path description.

Worboys *et al.* [8], [15] further investigated the methods to model indoor space and generate the navigation path from the floor plan by classifying the nodes into different types in very comprehensive manner. The types of nodes –room node, portal node, stairwell node, elevator node, landmark node, mid-path node, terminal node, turning node, junction node– contain rich semantic labels.

2) Floor plan analysis for robots navigation: Floor plans were directly used as the metric map for robot navigation.

Setalaphruk *et al.* proposed a method for robot navigation from a sketch floor plan in [18]. The method assumed the corridors environment without metric information. Augmented topological map then was generated from sketch floor plan by identifying structure of the corridors. Multiple hypotheses of the robot's localization were maintained and updated during navigation in order to cope with sensor aliasing and landmark-matching failures due to factors such as unknown obstacles inside the corridors.

Gao *et al.* proposed a deep learning based method: intention-net, to perform the robot indoor navigation in [19]. The method focused on the learning of the low level controller from vision features so that it can tolerate some localization errors. On the high level, the floor plan was used to conduct localization and path planning.

The above-mentioned works shared the similar assumption that the floor plans were relatively accurate representations of the environment. This might be true for most corridor environments. But for general indoor space, there might be large discrepancies between floor plans and on-ground reality due to the placement of furniture, cubicles, dynamic objects etc. The localization errors in such situations might not be properly handled by above methods.

Boniardi *et al.* proposed vision based robot localization using floor plans by matching the edge-based room layout in [20]. The methods relied on edge extraction and it might not work well if the camera was facing a wall inside a room. It was also not clear how to use the localization methods for path planning and navigation.

B. Topometric map based navigation

Thrun proposed a metric-topological based map for robot navigation in [5]. In his work, the metric map was generated by the exploration of the environment. The topological map was extracted from the metric map by generation of Voronoi Graph. The topological map was then used for fast path planning while the metric map was used for localization.

Bazeille *et al.* proposed a vision based topometric map for qualitative robot localization in [6]. The method built the topometric map by tele-operations and collected the images along the traversed path. During navigation, the images obtained from the sensors were compared to the collected images to determine the rough pose of the robot.

Similarly, Oliverira proposed a deep learning based topometric localization in [7], where the LIDAR based localization information as groundtruth were encoded into the neural network during training. During its traversal, a robot's location is predicted by the neural network from its observation. The method was more similar to the path following by tracking landmarks along the path.

Mazuran *et al.* proposed a method to represent the whole environment in a topological format while retaining the metric information to the local regions in [4]. Thus the requirement for global map consistency can be relaxed. Tully *et al.* proposed a similar method in [21] to divide the whole environment into different regions as the topological map and metric information was represented as an atlas.

Cheng *et al.* proposed indoor topological localization and navigation in [22]. They assumed that the indoor environment consists of rooms and corridors, and focused on how to automatically build the topological map for robot navigation in unfamiliar scenes. The main problems they solved were corridor type classification and loop closure detection.

Our topometric map is different from the other above-mentioned methods in the following aspects:

1) The distribution of nodes is sparse in our topometric map;
2) It does not provide metric features for accurate localization everywhere;
3) It does not treat the floor plan as the accurate metric map for localization.
4) It does not assume corridor-like environments;
5) It forms the complete navigation package together with the additional novel path planning method.

III. The Proposed Cognitive Navigation Package

The proposed package includes two important components: 1) Generation of the topometric map; and 2) Navigation via the topometric map. The details are given in this section.

In all the following algorithms, superscripts $\{^l, ^s, ^f\}$ indicate the metric information with respect to the *l*aser scan map centered at a robot, the *S*LAM map between two nodes and the *f*loor plan respectively.

A. Generation of the topometric map

Our topometric map is represented as $\{N, E\}$ as shown in table I, where N represents nodes and E represents edges.

Nodes are selected according to the following criterion:(1) there exist multiple paths leading to different directions ; (2) there is only one path available between any two connecting nodes. These criterion significantly reduce the number of nodes and ensure the conciseness of the topometric map. They also mandate that robot decisions on moving direction are made at nodes. To achieve this, nodes have visual features to allow a robot to recognize node locations, and to correct its pose for better decision making.

TABLE I

Elements of the topometric map

Elements	Contains	Remarks
$N = \{n_i\}$ (Nodes at Junctions)	$(ax_i^f, ay_i^f, a\theta_i^f)$	Anchor point position on floor plan
	r	Radius for visual features collection
	F_{vlad}	VLAD Features for place recognition
	F_{3d}	3D Features for pose recovery
$E = \{e_{ij}\}$ (Edges between Nodes)	$e_{ij} = 1$ if path exists btw n_i, n_j	Indicate two connecting nodes
	$G_{ij}^f = \{g_m\}$	Nominal path btw nodes n_i and n_j
	$g_m = (gx_m, gy_m, g\phi_{m+1,m}, g\phi_{m-1,m})$	Direction guide points on G_{ij}^f (gx_m, gy_m) location; $g\phi_{k,m}$: direction from g_m to g_k

Edges are defined in between two connecting nodes. Ideally, if the path between two connecting nodes is a straight line, the edge can be described as the relative position between two nodes in polar coordinate format (r, θ), where r, θ are the distance and direction between two nodes. But in order to handle the curved path, a nominal path is used to represent an edge in this work. A nominal path comprises of direction guide points, down-sampled from the shortest path between two connecting nodes. A nominal path is a *virtual* path on the free space of the floor plan, and is not directly used for path planning. The guide points $\{g^{ij}\}$ are used to describe the potential direction changes along an edge e_{ij} between two nodes $\{n_i, n_j\}$.

Algorithm 1 Generation of the topometric map

Input:
The floor plan
Output:
The topometric map
Floor plan analysis:
  Extraction of the topological map $\{N, E\}$
  Detection of anchor points $\{ax_i^f, ay_i^f, a^f\theta_i\}$
for each $n_i \in N$ **do**
  Verify n_i;
  Associate $n_i \leftarrow$ its nearest $\{ax_i^f, ay_i^f, a\theta_i^f\}$
  Detect the corner $(cx_i^l, cy_i^l, c\theta_i^l)$ from sensor data
  Associate $(cx_i^l, cy_i^l, c\theta_i^l) \leftrightarrow (ax_i^f, ay_i^f, a\theta_i^f)$
  Get F_{vlad} and F_{3d} within the range r of n_i
end for
for each $e_{ij} \in E$ **do**
  Compute the shortest A^* path $\{SP\}_{ij}^l$ between
    the two nodes n_i and n_j
  Compute nominal path G_{ij}^l by downsampling
    the points along $\{SP\}_{ij}^l$
end for
return The topometric map as presented in table 1

To build the topometric map, the floor plan is taken as the initial map with two assumptions: (1) it is drawn to scale; (2) missing and wrong components are allowed, but the latter should ideally be minimal. Most floor plans should satisfy these two assumptions. By using of floor plans, exploration of the whole environment can be avoided as long as the local coordinate system can be established. The anchor points in table I are served for that purpose. They typically refer to the salient structural corners near nodes. Their correspondents in the real environment should be easily detected and verified.

Algorithm 1 shows the steps in generating a topometric map: (1) extraction of the topological map from floor plans; (2) verification by observation of environment; (3) updating with metric information; (4) generating direction guide points along nominal paths.

Here are some added remarks:

- *Nodes candidates selection*: To meet the criterion of nodes, we put nodes on junctions where robots need to make decisions on a change of moving direction. Junctions are key traversal locations and are expected to be rarely blocked over a long time.
- *Additional sub-nodes selection*: In the real environment, the distance between two nodes could be quite long. Coarse localization based on odom or SLAM might not be good enough, especially when the nominal path is not a straight line. In this case, additional sub-nodes can be added between two junctions.
- *Visual features association*: To facilitate the functional role of nodes (for localization) and to cope with floor plan discrepancies, the topological map is updated with metric information. Two kinds of visual features are extracted, including (1) key-points based VLAD descriptors [23] for place recognition; (2) key-points based 3D reconstruction [24] for pose recovery .
- *Manual initial map correction*: The floor plan discrepancies may make the construction of visual nodes difficult. Hence, some manual efforts may be needed to drive the robot to each junction to verify the anchor points association. Note: this is still more efficient than the full metric map building since it is done only at the nodes, and without the need for compute intensive loop closure and global bundle adjustment.

We show an example to illustrate the topometric generation in Figure 1. The floor plan of our office building is used. In the topometric map we computed, only 10 places are selected as nodes, each roughly covering a circular area of radius $3-5$m. The nominal paths are represented by the direction guide points (green dots). For better visualization, only a subset of the latter is shown.

B. Navigation with the topometric map

Because metric information for localization is sparsely confined only to node regions, the proposed topometric map is incompatible with the traditional two level path planning framework for robot navigation. Instead, in Algorithm 2, we propose a direction based navigation algorithm that is more suitable for topometric maps. We shall see that the underlying steps seek to mimic the way humans navigate in the following two fundamental aspects:

Localization: A hybrid state is adopted to localize the robot. The hybrid state includes one discrete state to specify the previous node and a continuous state to indicate the current location with respect to the previous node. In other words, the previous node is a reference frame for the localization on the corresponding edge [11]. On nodes, the accurate localization is performed via visual features. On edges, either SLAM or odom can be used to estimate the distance from the previous node. The Odom is efficient but less accurate and the SLAM needs more computation resource but it is more accurate. With the proposed hybrid state, all the submaps of the edges can be more independently if the SLAM is selected to estimate the distance and they are not required to be globally consistent.

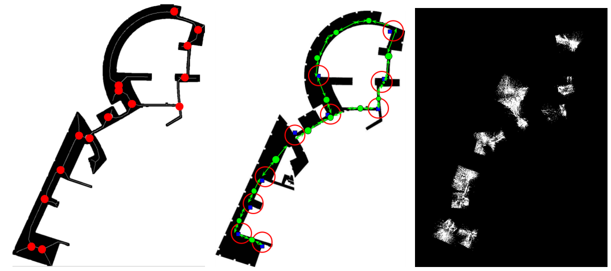

Fig. 1. Illustration of generation of topometric map: Left: an initial topological map with nodes in red and edges in blue; Middle: the verified topological map with nodes in red, direction guide points in green and anchor points in blue; Right: 3D features associated to nodes

Path planning: Addresses the question of what is the optimal or near optimal path given a start point and a goal point. There are three levels of path planning:

Global level: Global path planner computes a *logical* path from the given start position to the given goal position using the topological ordering of the nodes. This is done efficiently since there is no need for any sensor observation input. The global path planner breaks down the global path into pieces (regional level) according to the orders of the nodes along the logical path.

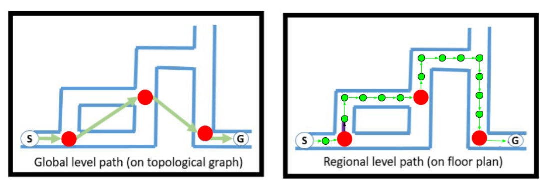

Fig. 2. Explanation of global level logical path and regional level nominal path. Left: the logical path shown as green arrows; Right: the nominal paths shown as a group of green dots

Regional level: Regional path planner similarly computes a nominal path from given start node to the next node on the global path without the need for any sensor data. The global nominal path can be obtained by combining all regional nominal paths along the logical path. Note: for efficiency, the nominal paths between any two connecting nodes are pre-computed once from the free space on the floor plan. The nominal path is used to guide a robot to determine the next way point from the sensor reading for local path planner.

A simple illustration is shown in Fig. 2 to explain the difference between the global path and the regional path.

Local level: Local path planner emphasizes on the collision

Algorithm 2 Navigation algorithm

Input:
Start Pose at Node n_{st}
End pose at Node n_{ed}
The topometric map $\{N, E\}$
Sensor Data
Output:
Success if reaching at Node n_{ed}
Fail or else

Initial current pose as slam origin S^f at node n_{st};
Compute Logical path LP via $\{N, E\}$
Set $n_{prev} = n_{st}$
while $n_{prev} \neq n_{ed}$ **do**
  Get n_{next} on LP
  Get nominal path $G \in [n_{prev}, n_{next}]$
  Initialize SLAM Map $SM = \emptyset$
  Initialize Robot location on SLAM Map $\hat{R}^s$
  Obtain guide point $g_{cur} = g_0$ on G
  while $S^f + \hat{R}^s \notin n_{next}$ **do**
    while $S^f + \hat{R}^s \leq g_{next}$ **do**
      Obtain sensor data
      Extract region exit points $\{e_i^l\}$;
      Select best exit e_{best}^l as next way-point;
      Start local movement to e_{best}^l
      Update robot position $\hat{R}^s += e_{best}^l$
      Update SLAM Map
    end while
    Set $g_{cur} = g_{next}$
  end while
  Set $n_{prev} = n_{next}$
  Update current robot pose as S^f at node n_{next}
end while

free movement to the next way-point from the actual sensor data. It is based on the egocentric representation crucial for navigating in confined and dynamic environments. Unlike its local path planner counterpart for the metric map based navigation, it is very challenging for a topometric based local path planner to determine the next way-point. In the next subsection III-C, we discuss a novel directional driven approach to do that.

C. Direction guide next way-point detection for local level path planning

The next way-point algorithm is determined based on the fact that there are only limited potential exits to allow the robot leaving the current space. These potential exits normally refer to the traversable space. Corridors are examples of such well-structured space with only two exits pointing to opposite directions. However, other spaces may not be as well-formed, e.g. between cubicles, chairs, etc. There might be many broken "exits" along the main direction. Although the coarse localization is not reliable for path planing, the relative direction from the current coarse location to the goal is by and large accurate. This can be used to select the best exit point as the next way-point for local motion planning.

Algorithm 3 Detection of Region Exit Points

Input:
$\mathbf{P}^l = \{p_i^l = (p_{r,i}^l, p_{\phi,i}^l)\}$: sensors reading;
Output:
$\mathbf{E}^l = \{e_i^l = (er_i^l, e\phi_i^l, e\phi_n^l)\}$

Euclidean clustering of $\mathbf{P}^l$ into N pieces $\{(ps_n^l, pe_n^l)\}$
  where ps_n^l and pe_n^l are two endpoints of the n^{th} piece
for each $(ps_n^l, pe_n^l), where \quad n < N$ **do**
  $dist = \inf$; $index = -1$
  if $ps_n^l > pe_n^l$ **then**
    $P_{tmp} = pe_n^l$
  else
    $P_{tmp} = ps_n^l$
  end if
  for each $p_i^l \in \mathbf{P}^l$ **do**
    if $p_i^l \in n$ **then**
      $dist_1 = |P_{tmp} - p_i^l|$
      if $dist_1 < dist$ **then**
        $index = i$; $dist = dist_1$
      end if
    end if
  end for
  $e_n^l =$ the mid point between p_{tmp} and p_{index}
  $e\phi_n^l = \pi/2+$ slope of line (p_{tmp}, p_{index})
end for
return $\{e_n^l\}$

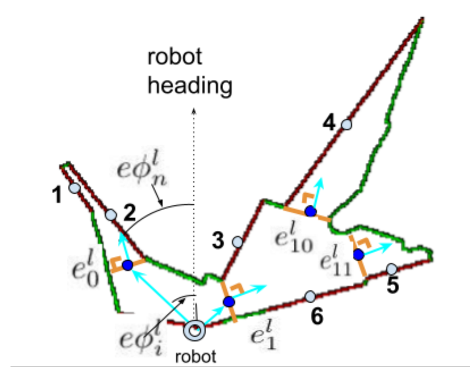

Fig. 3. Region Exit points detection. Green lines: sensor reading; Red Lines: frontiers; White dots: Mid Points of frontiers; Dark Yellow Lines: Narrowest paths towards frontiers; Blue dots: exit points on the narrowest paths towards frontiers; Blue lines: normal direction of exit points

Detection of region exit points is elaborated in Algorithm 3. From sensor's observation, the frontiers are commonly used to indicate the potential boundaries towards unknown space [25]. The exit points are based on frontiers. An exit point is defined as the midpoint of the narrowest path (exit line) towards frontiers. One exit line implies the only way towards other regions. The exit point is described as $\mathbf{E}^l = \{e_i^l = (er_i^l, e\phi_i^l, e\phi_n^l)\}$, where $(er_i^l, e\phi_i^l)$ is the polar coordinate of an exit point; and $e\phi_n^l$ is the potential path direction the exit point is leading to and is perpendicular to the exit line.

In Fig. 3, frontiers $1, 2$ have the same exit point e_0^l. Frontiers $3-6$ have the same exit point e_1^l. e_{10} and e_{11} are helpful to decide whether e_1 is a better choice. Although there are 6 frontiers, only two directions are available to move at the moment.

Algorithm 4 Best region exit point detection from sensor data

Input:

$\hat{\mathbf{R}}^s = (\hat{r}_r^s, \hat{r}_\phi^s)$: Robot pose on SLAM map

$\mathbf{S}^f = (s_r^f, s_\phi^f)$: SLAM map origin on floor plan

$\mathbf{E}^l = \{e_i^l = (er_i^l, e\phi_i^l, e\phi_n^l)\}$: Region exit points

$g_m^f = (gx_m^f, gy_m^f, g\theta_{m-1,m}^f, g\theta_{m+1,m}^f)$: Next guide point on floor plan;

Output:

The best exit point for local motion planning

for each $e_i^s \in \mathbf{E}^s$ **do**

$\quad \Delta\alpha_1{}_i^f = |e\phi_i^l + \hat{r}_\phi^s + s_\phi^f - g\theta_{m-1,m}^f| \qquad (A.4.1)$

$\quad \Delta\alpha_2{}_i^f = |e\phi_n^l + \hat{r}_\phi^s + s_\phi^f - g\theta_{m+1,m}^f| \qquad (A.4.2)$

$\quad f(e_i) = \Delta\alpha_1{}_i^f + \Delta\alpha_2{}_i^f \qquad (A.4.3)$

end for

return $\text{argmin}_{e_i} f(e_i)$

The steps to determine the best region exit point is described in Algorithm 4, where $\Delta\alpha_1{}_i^f$ estimates how close it is between the direction towards an exit point and the direction towards the next guide point; $\Delta\alpha_2{}_i^f$ estimates how close it is between the direction an exit point leading to and the direction the next guide point leading to; $f(e_i)$ is the cost function by combining these two directions difference. The e_i with the smallest value of $f(e_i)$ is the best region exit point towards the next node.

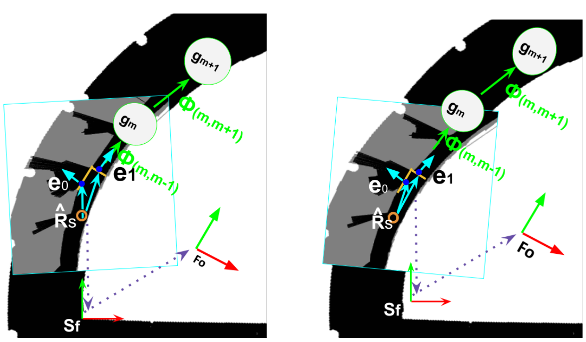

Fig. 4. An example of selecting the best region exit points. Left and right are snapshots of two navigation runs on the same environment. Note the different robot poses due to localization errors. Notwithstanding, the correct e_1 exit point is selected in both cases, showing the robustness of the algorithm to localization errors. See text for details. $\hat{R}_s$: Robot frame; S_f: local SLAM frame; F_O: floor plan frame; Purple lines indicate TF

In Fig. 4, we show an example of determination of the best region exit point from a direction guide point. The only difference between the left and right images is the robot pose $\hat{R}^s$ on the floor plan. This could have arisen due to errors from coarse localization. Notwithstanding, the region exit points e_0 and e_1 are observable from both robots. The direction guide points g_m and g_{m+1} are also shown in Fig. 4. It is clear that the exit point e_1 from both images has the better direction similarity comparing to the guide point g_m than the exit point e_0 has. In this example, both robots correctly move to exit point e_1. Since the local planning is based on selecting the best region exit point, it is not sensitive to the robot coarse localization $\hat{R}^s$.

Once the next way-point is determined, the existing local path planner such as the E-band or the DWA can be adopted for collision free movement. It should be pointed out that both the E-band and the DWA can be improved by taking trajectory of each moving obstacle in a neighborhood as well as asymmetry between free-space in front of the moving obstacle and free space behind the moving obstacle into consideration. As such, the robot will always try to utilize the free space behind the moving obstacle.

IV. Experiments and Discussions

Due to the fact that little work used floor plans to perform the navigation without instant precise localization, the following experiments are conducted to show how our proposed system utilize floor plan, which contains considerable discrepancies, to perform navigation with the help of limited metric information from selected regions.

A. Experiments

Hardwares: In experiments, Pioneer P3-DX mobile robot is used with additional sensors of 1. Hokuyo 2d LIDAR for anchor points and region exit points detection; 2. one webcam for visual features extraction.

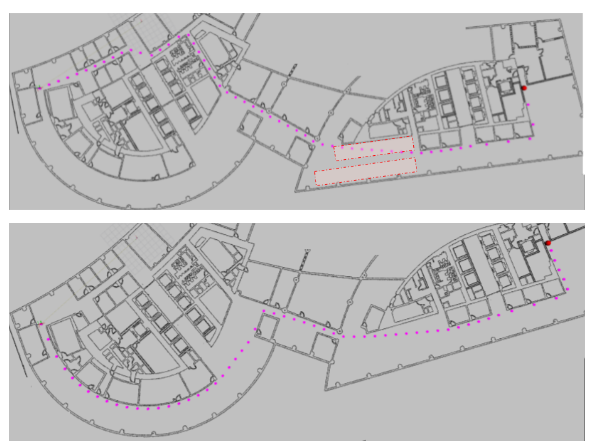

Fig. 5. Two test routes. route 1: indicated by pink dots on the top image; route 2: indicated by pink dots on the bottom image. The most difficult areas are marked as dashed rectangles. The cubicles are missing from both sides of the real path.

Environments: The test environments are located in our office building as shown in Fig. 1. Prepossessing is applied on the floor plan to keep the main structure and ignore the rooms along the traversal paths. The two paths used in the environments are shown in Fig. 5. The distance for both routes is about 150 meters.

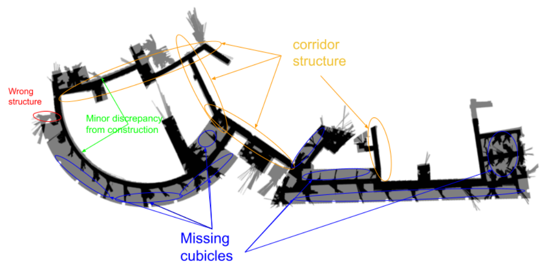

Fig. 6. Discrepancies between floor plan and 2D SLAM Map. Dark pixels: the SLAM map, non-white pixels: the floor plan

Discrepancy: For comparison, we build a SLAM map using gmapping provided by ROS package [26]. The discrepancy is shown in Fig. 6, where black pixels show free space on the SLAM map and non-white pixels show free space on the floor plan. Here we do not have an assumption of corridor environment like the other floor plan based methods. There are a lot of cubicle space missing from the floor plan. In addition, there exist some minor discrepancy from the construction between the floor plan and the real environment. This could affect the performance if a floor plan is directly used as the metric map. Despite discrepancy, the topological relationship from both maps are quite consistent.

Some parameters used: 1. The radius of a node $r = 3\ 5$ meters; 2. The distance between direction guide point on the nominal path is 1 meter; 3. 10 nodes in total are selected and 2 more sub-nodes are inserted along the curve arcs. 4. Cartographer [27] is used to maintain the SLAM map for the navigation between two nodes.

Experimental results: To show the effectiveness of our proposed methods, we conducted the following experiments:

1) navigation with the relative start/goal positions without any map
2) navigation with the floor plan as metric map
3) navigation with the proposed topometric map

The navigation method in [28] is used in experiment (1). It is in fact the exploration towards a specific goal. By exploration itself, it is able to build a SLAM map. But it is very sensitive to the initial relative pose between starting point and goal point. Furthermore, its drift gets bigger and bigger as the distance increases. As there is no prior information, it is difficult to determine if the goal is reached. It is also very time consuming as it will always try every possible frontiers.

For experiments (2) and (3), each was ran 10 times (5 times for route 1 and 5 times for route 2). We show the most difficult parts for topometric map based methods in Fig. 7 and compare it with the other methods.

The floor plan based navigation always fails when the localization is not possible due to the large discrepancy as indicated in the upper image of Fig. 7. Navigations based on our topometric map are all successful. More details please refer to the supplement video or visit `https://youtu.be/1Z1NedDtmao`.

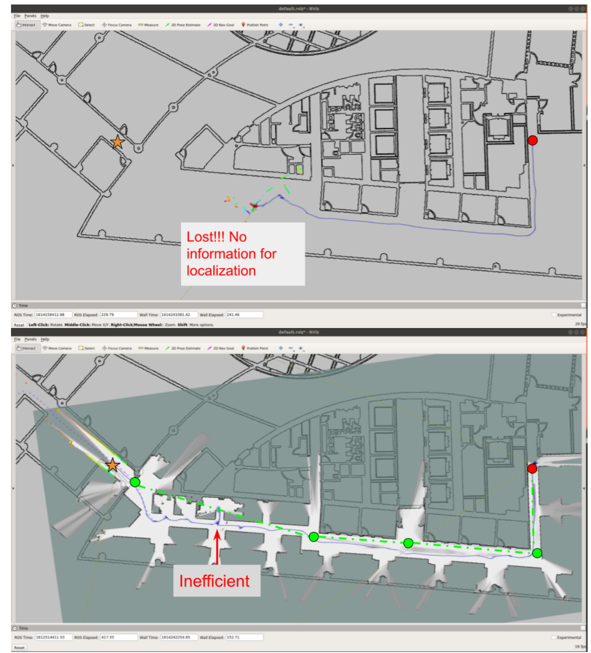

Fig. 7. Comparisons between Floor Plan(FP) based (upper) and proposed topometric based navigation(lower). Start from the red circle at right to the five star at left. Green circles: nodes; green dashed lines: nominal path.

Due to the discrepancy, the efficiency of our proposed method is slightly worse as shown in the lower image of Fig. 7, in which the robot enters into the cubicle area. But considering our method does not require precise localization everywhere, there are still some advantages compared to the metric map based methods. We shall explain in the next section IV-B.

B. Discussions

Importance of the nodes: If A node fails to be recognized and used to correct the drift, our method will fail. In this case, we accumulate the evidence over a sequence of observations within the node radius to make robust node recognition and pose correction.

Cost for prior mapping and maintenance: we use floor plan as prior map and update the visual features locally on nodes at junctions. The number of junctions is normally small. This will significantly reduce the cost of prior mapping and corresponding maintenance. In addition, it is easily to find the anchor point to build the visual map locally, so loop closure and global bundle adjustment are not required.

Error corrections: In our topometric map based navigation, the mechanisms for error corrections are provided to avoid localization lost by place recognition and reduce drift by pose correction. For the metric map based methods, these are treated as the different tasks and need special handling.

Alternative to the existing map: Our proposed method is a viable alternative to existing methods. It has the potential to handle the case when precise localization is not possible due to lack of features or crowded scene.

Compatible to human knowledge: With increasing and pervasive use of service robotics, human-robot interaction

(HRI) during navigation will be critical. Our topometric map representation is compatible with how humans would perceive the environment. Here is an example of a useful HRI in action: a robot is lost, and asks a human for help; the human responds by giving some directional verbal instructions; the robot uses them to find its way to goal.

V. CONCLUSIONS AND FUTURE WORKS

A novel cognitive navigation package has been introduced on top of floor plan. The proposed method provides one alternative to existing accurate prior map based methods, especially when the precise localization is not possible due to lack of features or crowded scene. The proposed package includes a unique component for the robot to achieve the directional based navigation. The experiments demonstrate the capabilities of our proposed framework.

Our current package has some limitations. For example, the current implementation may still need some manual effort for the first traversal through the real environment. It can be improved by automatically updating the floor plan by adding on the visual features and integrating with semantic information. The improved version can be adopted to achieve very accurate point-to-point navigation together with the visual inertial odometry in [29] and the controller in [30]. It is also important to investigate how to deal with the complex environment rather than in normal office environment. All these works will be studied in our future research.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2015.

[3] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "Inloc: Indoor visual localization with dense matching and view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7199–7209.

[4] M. Mazuran, F. Boniardi, W. Burgard, and G. D. Tipaldi, "Relative topometric localization in globally inconsistent maps," in *Robotics Research*. Springer, 2018, pp. 435–451.

[5] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.

[6] S. Bazeille and D. Filliat, "Incremental topo-metric slam using vision and robot odometry," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4067–4073.

[7] G. L. Oliveira, N. Radwan, W. Burgard, and T. Brox, "Topometric localization with deep learning," in *Robotics Research*. Springer, 2020, pp. 505–520.

[8] L. Yang and M. Worboys, "Generation of navigation graphs for indoor space," *International Journal of Geographical Information Science*, vol. 29, no. 10, pp. 1737–1756, 2015.

[9] X. Zang, A. Pokle, M. Vázquez, K. Chen, J. C. Niebles, A. Soto, and S. Savarese, "Translating navigation instructions in natural language to a high-level plan for behavioral robot navigation," *arXiv preprint arXiv:1810.00663*, 2018.

[10] G. Sriharee, "A symbolic-based indoor navigation system with direction-based navigation instruction," *Procedia Computer Science*, vol. 52, pp. 647–653, 2015.

[11] T. Wolbers and J. M. Wiener, "Challenges for identifying the neural mechanisms that support spatial navigation: the impact of spatial scale," *Frontiers in human neuroscience*, vol. 8, p. 571, 2014.

[12] A. D. Ekstrom and E. A. Isham, "Human spatial navigation: representations across dimensions and scales," *Current opinion in behavioral sciences*, vol. 17, pp. 84–89, 2017.

[13] P.-Y. Gilliéron and B. Merminod, "Personal navigation system for indoor applications," in *Proceedings of the 11th IAIN world congress*, no. CONF, 2003.

[14] B. Lorenz, H. J. Ohlbach, and E.-P. Stoffel, "A hybrid spatial model for representing indoor environments," in *International Symposium on Web and Wireless Geographical Information Systems*. Springer, 2006, pp. 102–112.

[15] M. Worboys, "Modeling indoor space," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, 2011, pp. 1–6.

[16] M. Xu, S. Wei, and S. Zlatanova, "An indoor navigation approach considering obstacles and space subdivision of 2d plan," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 41, p. 339, 2016.

[17] F. Mortari, E. Clementini, S. Zlatanova, and L. Liu, "An indoor navigation model and its network extraction," *Applied Geomatics*, vol. 11, no. 4, pp. 413–427, 2019.

[18] V. Setalaphruk, A. Ueno, I. Kume, Y. Kono, and M. Kidode, "Robot navigation in corridor environments using a sketch floor map," in *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No. 03EX694)*, vol. 2. IEEE, 2003, pp. 552–557.

[19] W. Gao, D. Hsu, W. S. Lee, S. Shen, and K. Subramanian, "Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation," *arXiv preprint arXiv:1710.05627*, 2017.

[20] F. Boniardi, A. Valada, R. Mohan, T. Caselitz, and W. Burgard, "Robot localization in floor plans using a room layout edge extraction network," *arXiv preprint arXiv:1903.01804*, 2019.

[21] S. Tully, H. Moon, D. Morales, G. Kantor, and H. Choset, "Hybrid localization using the hierarchical atlas," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 2857–2864.

[22] H. Cheng, H. Chen, and Y. Liu, "Topological indoor localization and navigation for autonomous mobile robot," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 729–738, 2014.

[23] R. Arandjelovic and A. Zisserman, "All about vlad," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 1578–1585.

[24] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[25] B. 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'*. IEEE, 1997, pp. 146–151.

[26] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[27] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1271–1278.

[28] G. Haddeler, J. Chan, Y. You, S. Verma, A. H. Adiwahono, and C. Meng Chew, "Explore bravely: Wheeled-legged robots traverse in unknown rough environment," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 7521–7526.

[29] J. Henawy, Z. Li, W. Y. Yau, and G. Seet, "Accurate imu factor using switched linear systems for vio," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7199–7208, 2021.

[30] Z. Li, W. Gao, C. Goh, M. Yuan, E. K. Teoh, and Q. Ren, "Asymptotic stabilization of nonholonomic robots leveraging singularity," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 41–48, 2018.