Semantic enhanced navigation among movable obstacles in the home environment

Nico Sun, Erfu Yang, Jonathan Corney, Yi Chen and Zeli Ma

Abstract— An autonomous household robot has to be able to navigate through a variable environment and perform common household tasks. Additionally, in cluttered and narrow homes movement can become impossible unless obstacles are moved out of the way. Both challenges involve the manipulation of objects and a planning algorithm that can integrate the function of regions and objects to avoid the creation of new safety hazards during robot movement. We present a semantic detection method during path planning using a gridded semantic map to improve navigation among movable obstacles (NAMO) and support for simple household sub-tasks like cleaning a table and moving obstructing objects to another location. In our tests, the spatial planning was completed well within human reaction time, which is important for a natural interaction between a human and a robot.

I. INTRODUCTION

Today's mobile robots navigate on a binary map, often scanned using Simultaneous Localization and Mapping (SLAM), dividing the workspace into free space and fixed obstacles. Some algorithms explored Navigation Among Movable Obstacles (NAMO), creating a ternary map (fixed obstacles, movable obstacles and free space). But robots operating in a human environment need to have a more complex understanding of their environment for autonomous navigation due to random temporary obstacles being placed in their way (e.g. chairs, bags) and it is frequently not possible to re-plan a new path (e.g. apartments with only one corridor).

Humans can easily identify what obstacles are movable and require the least effort to clear a path. However, obstacles are not always moved to a position which would require the least amount of effort, because this position would block another path which would need clearing at another time e.g. a doorway or a hazardous location (like right behind a corner). While a corridor. traditionally is empty space in navigation it isn't suitable to place an obstacle there because other people need to move through it. Perhaps the most dramatic example is a fire escape. This space needs to be encoded as free to move through, but not free to leave obstacles in it.

The main focus in robotic navigation has been getting from point a to point b. Rather than moving to a specific x/y coordinate humans move to a region or an object which have a dimension and multiple adjacent points as a valid goal location. Navigational planners could emulate this behaviour by checking a semantically annotated map.

In this paper, we use a semantic encoded map to improve the NAMO quality by considering the functions of space. The remainder of this paper is structured as follows: Section II gives an overview of related work in navigation and semantic mapping. Section III describes our semantic detection for navigational planning. Section IV presents our experimental results. Section V discussion and finally we conclude our findings in Section VI.

II. RELATED WORK

There exists an extensive literature on robot navigation however given the compactness of this paper we discuss only the two closest aspects.

A. Navigation Among Movable Obstacles

Existing NAMO algorithms can solve very complex environments with a large number of obstacles, but typically these environments are rooms with random obstacles and don't resemble a household or office. e.g. one room with 20 chairs, tables and a few sofas[1][2][3]. The obstacle placement decision is also devoid of any function of space or the blocking of other paths. Additionally, an execution time of a few seconds is perceived as too long for a typical human interaction.

Algorithms that divide the space into free connected space regions and occupied regions won't be useful for 3D maps because in 3D there will be only one free space region for most objects which could theoretically be moved along the ceiling. This, however, is not a practical approach to deal with obstacles.

B. Semantic Mapping

Semantic mapping has been developed to give sensor data a meaning similar to the human perception of the environment. Simple semantics exists as a topological map, circles representing rooms and lines connecting rooms without metric information [4]. A more direct encoding of semantics into a navigation map with bounding boxes has been done in [5]. By using an RGB-D camera and convolutional neural network (CNN) semantic mapping has been done pixel by pixel on a 3D point cloud map from a SLAM algorithm [6][7]. Further

N. Sun, E. Yang and J. Corney are with the Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow, G1 1XJ, UK. (e-mails: {nico.sun, erfu.yang, jonathan.corney}@ strath.ac.uk).

Yi Chen: Industry 4.0 Artificial Intelligence Laboratory, School of Computer Science and Network Security,Dongguan University of Technology, Dongguan 523808, China. (leo.chen@ieee.org)
Zeli Ma: Rofut (Shanghai) Technology Co.,Ltd
Shanghai 201800, China. (macius.ma@rofut.com)

research is needed to evaluate each semantic mapping method and their usefulness for autonomous navigation and household task performance.

In order to use the maps spatial semantic knowledge for complex household tasks, it needs to be combined with action-related and common-sense knowledge as in a semantic network [8][9]. The knowledge relates objects to another or to regions e.g. "microwave located in the kitchen" and their function e.g. "microwave can cook food".

III. OUR APPROACH

In order to encode different functions in a home environment, we utilize three layers for a 2D floor map, visualized as an RGB image with some predefined pixel by pixel encoded semantics. One layer for objects & obstacles, one for dynamic entities (humans, pets) and one for the room property or function.

For the robot pathfinding (start to goal) we use a bi-directional rapidly exploring random tree (Bi-RRT) algorithm (Figure 2) with a 5% goal bias. Bi-RRT is a variant of the simple RRT [10]. Simple RRT is great at exploring a large region of free space, but one single RRT can get stuck against a wall and have a hard time finding a gap like a doorway. In our tests, bi-directional RRT proved to be 3-10 times faster than a single tree, which is similar to findings in [11]. Due to the nature of a home robot environment – the interaction with humans. A solution should be found within human reaction time. Previous robots have been found to be too slow and unresponsive [12]. This gave preference of simple RRT over RRT*. After a solution is found we employ local path smoothing (Figure3) for a more natural motion.

The path planning and semantic detection (includes collision check) are done in Cartesian space instead of configuration space to avoids the recalculation of the space every time the robot moves an obstacle. The semantic detection of objects for the planner is done with OpenCV by finding the specified semantic value on the map and extracting its dimensions (contours). During path planning in the RRT algorithms, the semantic detection checks the map with the bounding box of the robot or movable obstacle and disregards a point when the bounding box touches another obstacle.

When the robot path is blocked by an obstacle the NAMO RRT, a simple modification to the RRT algorithm (see code) searches for a new collision free position that doesn't obstruct the robot's path or collide with other obstacles. The improvement of NAMO quality is done by excluding regions encoded on the semantic map as valid goal positions. This semantic check is only done after a new node is added to the RRT and not every time a node is checked against permanent obstacles. In our tests, for example, we excluded doorways as valid goal positions.

Algorithm 1 Semantic detection during RRT

```
 1: Map definitions
 2: Op = permanent obstacles
 3: Om = movable obstacles
 4: Rf = free region
 5: Rb = blocked region
 6:
 7: Robot navigation
 8: map = SemanticMap(Op,Om,Rf,Rb)
 9: goal = map(x,y)
10: while True do
11:     rnd = random(map)
12:     if rnd in Op then
13:         continue
14:     else
15:         add rnd to path
16:     if rnd in goal then
17:         break
18:     if smoothed path in Om then
19:         add Om to NAMO task
20:
21:     Movable obstacle goal check
22:     if rnd in goal then
23:         if rnd in Rb then
24:             continue
25:         else
26:             break
```

IV. TEST/EVALUATION

We perform the evaluation in Python 3.6 and single threaded on an Intel Core i5 2400 (3.1GHz). Ram usage is 60MB to load the 1000x800 three-layer test map + simple UI interface. During path planning, an additional 20MB is used for the computation of the robot path and new obstacle positions. The map resolution is 1cm/pixel, hence representing an apartment of 10m x 8m. The spatial semantic knowledge data is stored in NumPy arrays and visualized with matplotlib as an RGB image.

Pre-defined semantics and their visualization:

- Regions (red layer): unknown, sink, sinkstorage, TVcarpet, doorway, foodtable

- Dynamic Entities (green layer): unknown

- Objects (blue layer): unknown, plate, cutlery, TVtable,TVstool,plant,chair,foodtable, bed, shower

After an object is moved the map gets automatically updated (old position encoded as free space in object layer and new location encoded with the value of the object) For a better illustration of the skills we use the unused green layer

(dynamic entities) to enhance the contrast between the objects original and new position and dimensions.

As expected, checking a point against a list of semantic values takes longer than checking against a single value representing all obstacles. The time increase depends on the number of total semantic values. In our tests with 1000 known semantic values, the calculation time increased by a factor of 15-30. From 0.213s for one million single value to 3.1-6.7s for the same number of multiple value checks. So any path planning algorithm should still perform single value collision detection against unmovable obstacles to reduce the number of slower semantic checks.

A. Move to region or object
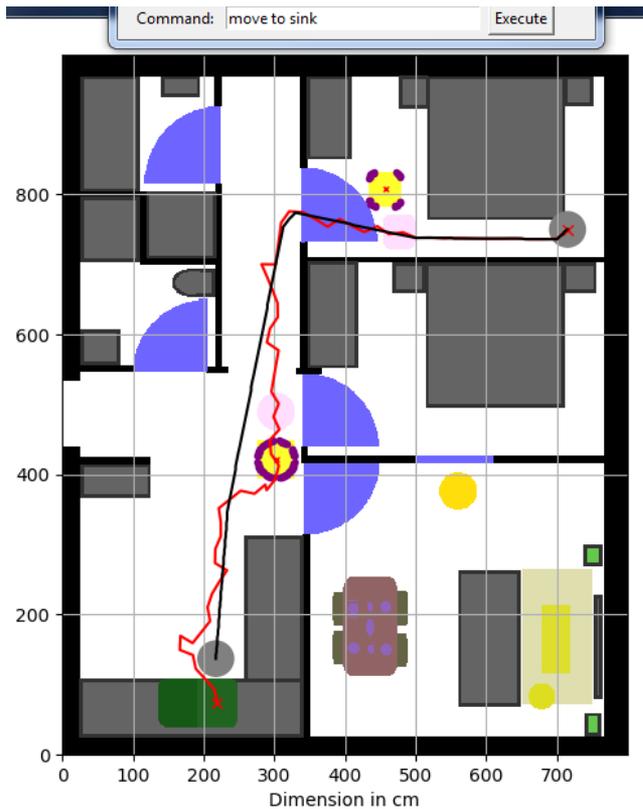
Figure 1: Robot skill: "Move to region/object" The robot (light grey) is in the upper right is instructed to move to the sink(green). The red line shows the raw path found by the Bi-RRT algorithm and the black line shows the smoothed path. In light red are the original obstacle positions & the purple dots outline the new positions for any moved obstacle.

Metric	Bi-RRT time in seconds	NAMO-RRT (seconds)	Robot Path cost	Obstacle path cost
Average	0.099	0.095	1049	199.0
Standard deviation	0.043	0.055	10.44	136.9

Table 1: Performance analysis for the task "move to sink". For our two-obstacle example the planning time for the obstacles is the same as for the robot path finding. While the robot path with a fixed goal region shows very little variations in length the obstacle movement shows high variation due to a flexible goal position.

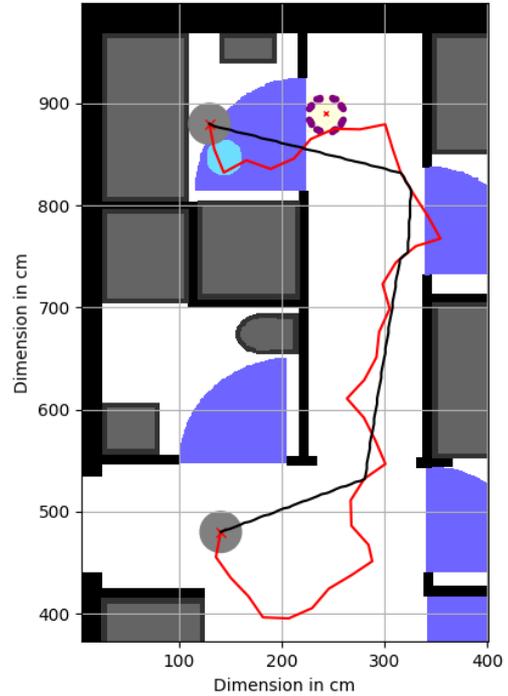### B. Movable obstacle placement of semantic NAMO algorithm

Figure 2: obstacle placement when considering semantics. The robot (light grey) is instructed to move from the lower left to the bathroom. The red line shows the raw path found by the Bi-RRT algorithm and the black line shows the smoothed path. The NAMO algorithm avoids placing the washing basket (light blue/purple dots) into the doorway (blue)
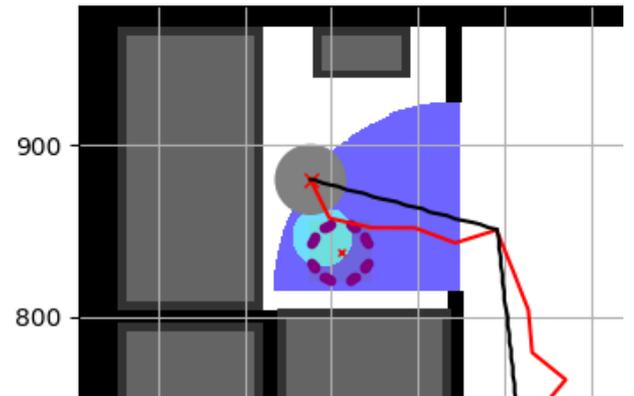

Figure 3: The same navigation task as in Figure 2 without exclusion of the door space(blue) as a valid goal position.

The obstacle movement cost (Table1) includes the cost of moving the robot and the relative size of the object compared to the robot. Small objects will have a negligible cost and large objects will have a high cost of moving. The performance analysis was done with 1000 samples in each test, a robot with a round base of 25 cm radius and an RRT expand distance of 20cm. For obstacles, the RRT expand distance is 10cm.

Equations for path cost:

$$C_R = d \qquad (1)$$
$$C_o = (1 + A_o/A_r) * d \qquad (2)$$

$C_R = Cost\ of\ moving\ robot, d = distance$
$C_o = Cost\ of\ moving\ obstacle, A = Area$

C. Moving objects to another region

When moving many small objects compared to the robot's size, the objects don't need additional collision detection. With the previously shown path cost calculation it's possible to calculate when it's more efficient to move the objects individually or to get a known container from a nearby place and move multiple objects at the same time. The semantically encoded map already includes the location and dimension of the goal region, therefore eliminating the need to compute a path for each individual object between its origin and goal position. Instead we only need to compute one path for the robot between the two regions and a short path for each object from the goal region to the objects final position.
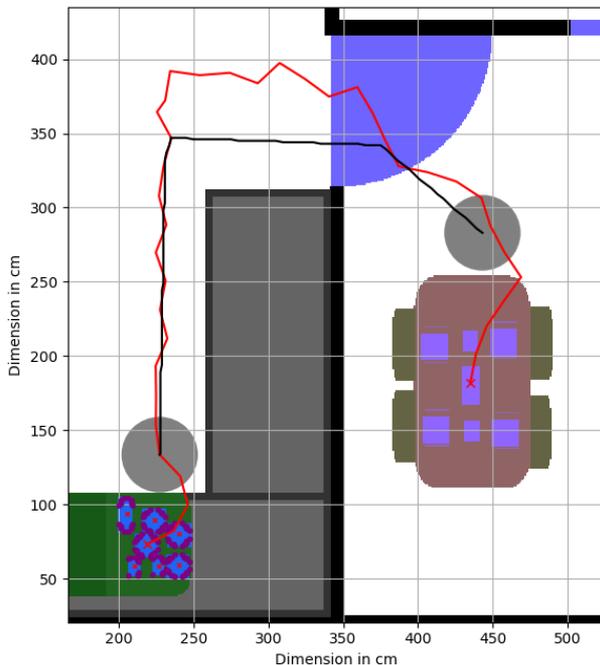

Figure 4: Robot skill: "move objects to a region". The robot moves all cutlery and plates (purple & purple dots) from the table(brown) to the sink(green). The robot path uses the same colours as before.

V. DISCUSSION & FUTURE DEVELOPMENT

The use of semantics in navigational planning isn't restricted to the RRT algorithm, for us the general nature of RRT allowed an easy combination of A to B navigation and spatial task planning for some household tasks.

A real household robot would greatly benefit from a 3D semantic map, especially for small stackable objects. The

performance of our currently un-optimized 2D representation was still well within human reaction time and an optimized version has the potential to work in 3D within reasonable human reaction time as well. Further evaluation needs to be done on a scanned semantic map instead of a pre-defined one.

VI. CONCLUSION

We have presented a semantic detection method during path planning for a gridded semantic map and how it can be used in a cluttered home with movable obstacles. With execution times of well below half a second, the semantics consideration can improve the navigation quality without adding significant computation time. By combining the planning for object placement and robot navigation into one system the system could also be used for practical household tasks, which are not yet well developed and needed in health care. In the future, these spatial planning tasks have to be combined with general knowledge of object functions and their usage/grasping to create household tasks that can be executed without specific prior knowledge of the exact environment.

REFERENCES

[1] M. Levihn, J. Scholz, and M. Stilman, "Hierarchical Decision Theoretic Planning for Navigation Among Movable Obstacles," Springer, Berlin, Heidelberg, 2013, pp. 19–35.

[2] M. Stilman and J. J. Kuffner, "NAVIGATION AMONG MOVABLE OBSTACLES: REAL-TIME REASONING IN COMPLEX ENVIRONMENTS."

[3] A. Akbari, Muhayyuddin, and J. Rosell, "Task planning using physics-based heuristics on manipulation actions," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–8.

[4] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3515–3522.

[5] H. Deeken, T. Wiemann, K. Lingemann, and J. Hertzberg, "SEMAP - a semantic environment mapping framework," in *2015 European Conference on Mobile Robots (ECMR)*, 2015, pp. 1–6.

[6] R. Li, D. Gu, Q. Liu, Z. Long, and H. Hu, "Semantic Scene Mapping with Spatio-temporal Deep Neural Network for Robotic Applications," *Cognit. Comput.*, vol. 10, no. 2, pp. 260–271, Apr. 2018.

[7] H. Sun, Z. Meng, and M. H. Ang, "Semantic mapping and semantics-boosted navigation with path creation on a mobile robot," in *2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2017, pp. 207–212.

[8] M. R. Petr Masek, "A Task Planner for Autonomous Mobile Robot Based on Semantic Network in Advances," *Adv. Intell. Syst. Comput. 393*, pp. 634–639, 2016.

[9] M. Tenorth, L. Kunze, D. Jain, and M. Beetz, "KNOWROB-MAP - knowledge-linked semantic object maps," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 430–435.

[10] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[11] A. H. Qureshi and Y. Ayaz, "Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments," *Rob. Auton. Syst.*, vol. 68, pp. 1–11, Jun. 2015.

[12] E. Martinez-Martin and A. P. del Pobil, "Personal Robot Assistants for Elderly Care: An Overview," Springer, Cham, 2018, pp. 77–91.