Semantic Path Planning for Indoor Navigation and Household Tasks

Nico Sun[1], Erfu Yang[1*], Jonathan Corney[1], and Yi Chen[2$]

[1] Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow , *corresponding author in UK
{nico.sun,erfu.yang,jonathan.corney}@strath.ac.uk
[2] Industry 4.0 Artificial Intelligence Laboratory, School of Computer Science and Network Security, Dongguan University of Technology, Dongguan 523808, China, $corresponding author in China leo.chen@ieee.org

Abstract. Assisting people with daily living tasks in their own homes with a robot requires a navigation through a cluttered and varying environment. Sometimes the only possible path would be blocked by an obstacle which needs to be moved away but not into other obstructing regions like the space required for opening a door. This paper presents semantic assisted path planning in which a gridded semantic map is used to improve navigation among movable obstacles (NAMO) and partially plan simple household tasks like cleaning a carpet or moving objects to another location. Semantic planning allows the execution of tasks expressed in human-like form instead of mathematical concepts like coordinates. In our numerical experiments, spatial planning was completed well within a typical human-human dialogue response time, allowing for an immediate response by the robot.

Keywords: Semantic path planning · Robotics · Semantic map · Navigation among movable obstacles.

1 Introduction

Todays autonomous mobile robots navigate on a binary map, often scan their surrounding environment using Simultaneous Localization and Mapping (SLAM) techniques [1], dividing the work-space into free space and fixed obstacles. Some algorithms explored Navigation Among Movable Obstacles (NAMO) [9], creating a ternary map (fixed obstacles, movable obstacles and free space). But robots operating in a human environment need to have a more comprehensive understanding of their complex 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 kept free.

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 close to an object which has a dimension and a region around them as a valid goal location. Navigational planners could emulate this behaviour by checking the dimensions of a region on a semantically annotated map.

This paper is an invited extension of an extended abstract presented at the UK-RAS 2019 [7]. The remainder of this paper is structured as follows: Section 2 gives an overview of the related work in navigation and semantic mapping. Section 3 describes our semantic detection method for navigational planning. Section 4 presents our experimental results. Section 5 discussion is made. Finally we conclude our findings in Section 6.

2 Semantic Path Planning

Semantics have been used in robotics to create a knowledge base to relate objects to other objects or to regions e.g. 'milk located in the fridge' and 'fridge located in the kitchen' [8, 12]. Objects can have a function as well e.g. 'fridge keeps milk fresh'. Using Semantics for path planning is still largely unexplored because only recently have SLAM algorithms with the help of neural networks been able to create a dense pixel by pixel encoded semantic map [5, 11]. On their own semantic slam algorithms still don't output the required quality for semantic path planning, but with a visualisation tool, a human user could correct misclassifications where needed.

With semantic path planning it's possible to detect during planning which region the algorithm is currently in. This means the state space isn't only divided into free space and obstacle space, it can have any identity such as 'on the carpet'. Combined with logical expressions such as 'if identity is' or 'if identity is not' a planner is able to avoid certain regions, never leave a specific region or stop planning when it reached a region. One usage example of this function is the manipulation of obstacles known as Navigation Among Movable Objects (NAMO). Unlike a road network, a home often has only one path to a goal eliminating the possibility of re-planning. If the path is obstructed by an obstacle a robot has to move it in order to clear the path. Existing NAMO algorithms can deal with numerous obstacles [9], but don't consider any function of space like the one required to open a door. This space can't be encoded as an obstacle because the robot has to move through it and it can't be encoded as free space because if an obstacle is placed into it it's impossible to open the door. Semantics can treat this space as free for the robot and free to move through with an obstacle, but not valid as a position to leave an obstacle.

Another usage of semantics path planning is for "Task and motion planning" (TMP). For household tasks moving objects from one region to another benefit from semantic segmentation of the search space. Small objects require a centimetre or even millimetre precision for grasping and when moving them to another room several meters away it's very inefficient to perform a multi-dimensional motion planning for the entire path at the grasping precision. Navigating between two different rooms only requires two dimensions with at most centimetre precision. The high dimensional and precise manipulator's motion planning to place an object only needs to be performed when the robot arm is within reach of the objects start or goal position. There has been previous work bringing objects to another location but didn't implement a division of the path into sub-paths with different resolutions and dimensions [2].

3 Methodology

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 and obstacles, one for dynamic entities (humans, pets) and one for the room property or function. For intelligent navigation, all semantic values are combined with a dictionary which contains the semantic map value, a keyword which is close human understanding and a property. Objects have the movable or unmovable property and regions have keepfree or usable for placing obstacles. The general format is shown in (1) and (2).

$$'object' = [semanticmapvalue, unmovable/movable] \tag{1}$$

$$'region' = [semanticmapvalue, keepfree/usable] \tag{2}$$

For the robot pathfinding (start to goal) we use a bi-directional rapidly-exploring random tree (Bi-RRT) algorithm with a 5% goal bias. Bi-RRT is a variant of the simple RRT [4]. Due to the nature of a home robot environment the interaction with humans. A solution should be found close to the typical response time of a human-human dialogue[10]. This requirement gave preference for simple RRT over RRT*. RRT* generates shorter paths but at a significant running time increase [3]. Previous robots have been found to be too slow and unresponsive [6]. After a solution is found we employ local path smoothing to reduce the path cost and for a more natural motion. 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). Thus all objects positions and dimensions are part of the map and not stored in a separate system. 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 is in contact with another obstacle. When the robot path is blocked by an obstacle the semantic NAMO RRT, a simple modification to the RRT algorithm searches for a new collision free position that doesnt

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.

In the first household task the scenario is: "clearing a region of objects". The semantic detection only allows the RRT algorithm to end if a node would place the object completely outside the specified region.

The second household task scenario: "moving objects to another region". The planner first checks if all objects fit into the goal region before attempting to find a path towards the region. After a new position for all specified objects was found in the goal region, a bi-directional RRT is calculated between the centre of the start region and centre of the goal region. During the path smoothing the planner returns the location adjacent to each region as the start and end position instead of the region centre.

4 Numerical Evaluation

We performed the evaluation in Python 3.6 and single threaded on an AMD 2700X. RAM usage was just 10MB for the 1000x800 three-layer test map. During path planning, an additional 10MB was used for the computation of the robot path and new obstacle positions. The map resolution was 1cm/pixel, hence representing an apartment of 10m x 8m. The spatial semantic knowledge data was stored in NumPy arrays and visualized with matplotlib as an RGB image. 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. In our tests with 1000 known semantic values, the calculation time increased by a factor of 15-30: from 0.213s for 1M 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.
The path cost is defined by:

$$C_r = d \tag{3}$$

$$C_o = (1 + A_o/A_r) * d \tag{4}$$

$$where, C_r = Costofmovingrobot, d = distance,$$
$$C_o = Costofmovingobstacle, A = Area$$

The obstacle movement cost includes the cost of moving the robot and the relative size of the object compared to the robot. Small objects will have a

Algorithm 1 Use of semantics during pathplanning

Map definitions
Op = permanent obstacles
Om = movable obstacles
Rf = keepfree regions
Ru = usable regions
map = SemanticMap(Op,Om,Rf,Ru)

Task: Move to region
goal(x,y) = SemanticMap(regionvalue)
find path
if smoothed path node in SemanticMap(regionvalue) **then**
    path end
if smoothed path node in SemanticMap(Om) **then**
    do Task move obstacles

Task: move obstacles
if path node in SemanticMap(Op) **then**
    discard node
if path node in robotpath or in SemanticMap(Rf) **then**
    continue search
else
    end search

Task: Clear a Region of Objects
if path node in SemanticMap(Op) **then**
    discard node
if node in SemanticMap(regionvalue) **then**
    continue search
else
    end search

Task: Moving objects to another region
startnode(x,y) = SemanticMap(regionvalue)
if path node in SemanticMap(regionvalue) **then**
    continue search
else
    discard node
if path node in SemanticMap(Om) **then**
    continue search
else
    end search
do Task Move to Region

negligible cost and large objects will have a high cost of moving. The performance metrics: running time and path cost have been calculated over 1000 sample runs for each test. Due to the random nature of the RRT algorithm the standard deviation of our results are quite high.

In the following subsections we show some common navigational tasks for a household.

4.1 Clear a Region of Objects

In order to vacuum a carpet, it first needs to be cleared of all obstacles. Otherwise, part of it remains dirty and develops discoloured edges around obstacles. The algorithm can exclude other regions as valid obstacle positions like the task in Fig. 3.

Table 1. Performance analysis for the task "Clear a region of objects (o)". t = time in ms, C = path cost

Metric	t_o	C_o
Average	61	263
Standard deviation	34	13

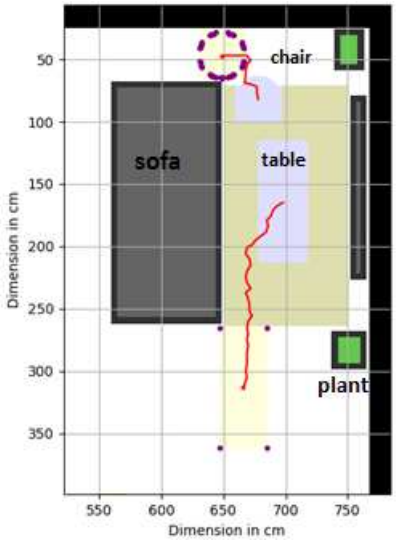

Fig. 1. Robot skill: clearing a region: Clearing a carpet(beige) of all movable obstacles, light blue: former obstacle positions, purple dots: new obstacle positions

4.2 Moving Objects to another Region

When moving many small objects compared to the robots size, the objects dont need additional collision detection. With the previously shown path cost calculation its possible to calculate when its 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, the algorithm only needs 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.

Fig. 2. Robot skill: move objects to a region. The robot moves all objects (purple dots) from the shelf (brown) in the bedroom to the sink(green) in the kitchen. The black line shows the smoothed robot path.

Table 2. Performance analysis for moving 5 objects from the shelf to the sink. t = time in ms

Metric	t_r	t_o
Average	112	8
Standard deviation	85	1

4.3 Move to a Region or an Object

As shown in Fig. 2 the robot is able to plan a path to another object encoded into the semantic map instead of fixed coordinates. When moving obstacles out of the way to clear the path the algorithm also considers the regions adjacent to the doors to still allow them to open and the robot to move through.

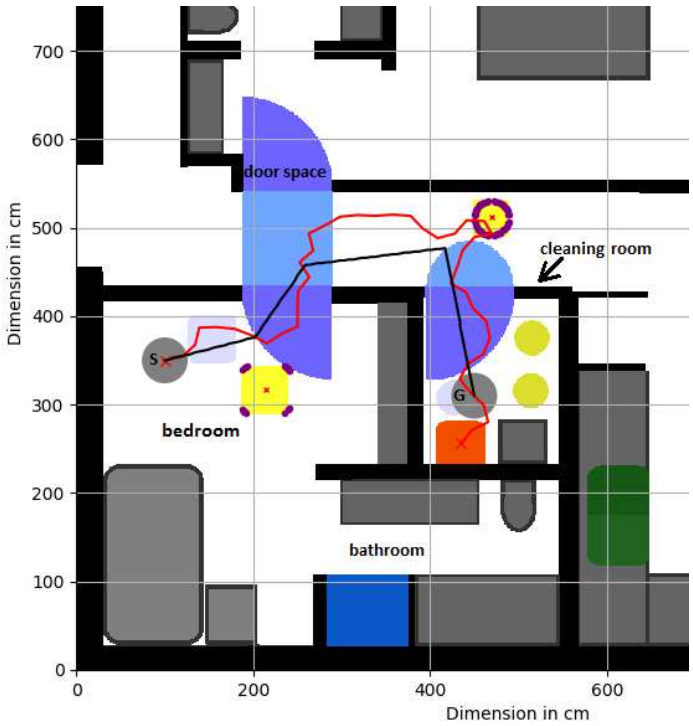

Fig. 3. Robot skill: Move to region/object The robot(grey) moves from the bedroom to the washing machine (orange) in the cleaning room. The red line shows the raw robot path and the black line shows the smoothed path. In light blue are the original obstacle positions and the purple dots outline the new positions for any moved obstacle.

Table 3. Performance analysis for the task "move to region". For our two-obstacle example the planning time for the obstacles (o) is the same as for the robot (r) path finding. t = time in ms, C = path cost

Metric	t_r	t_o	C_r	C_o
Average	82	90	580	364
Standard deviation	84	28	14	115

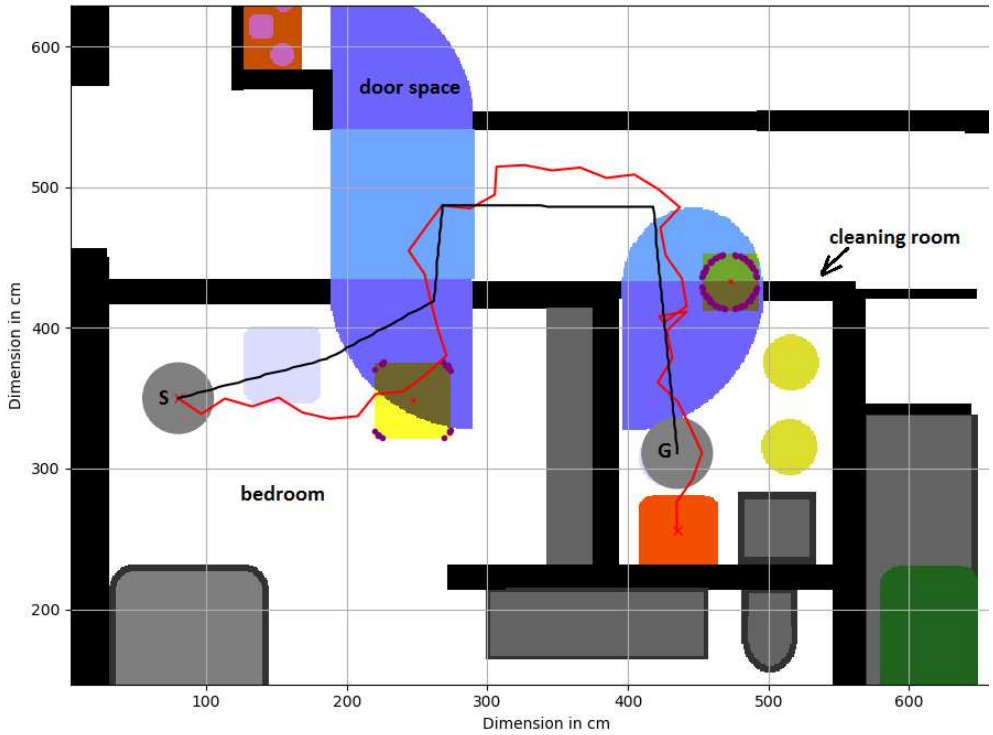

Fig. 4. Obstacle placement from task in Fig. 3 without considering semantics.

5 Discussion

Our test showed that consideration of semantics during path planning can enhance the navigation capability of robots. The pre-defined semantic map assumed a perfect semantic classification which isn't possible yet with existing semantic mapping methods. However, a simple user interface displaying the semantic map would allow a person to improve it by reducing noise and marking clear borders. A real household robot would greatly benefit from a 3D semantic map, especially when able to place small objects on top of others instead of only next to each other in two dimensions. The performance of our current 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 to ensure a desired quick response by a robot.

6 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 to avoid regions which should be kept free. With execution times of a tenth of a second in an apartment, 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 it solved partly the task and motion planning problem needed 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. Deeken, H., Wiemann, T., Lingemann, K., Hertzberg, J.: SEMAP - a semantic environment mapping framework. In: 2015 European Conference on Mobile Robots (ECMR). pp. 1–6. IEEE (sep 2015). https://doi.org/10.1109/ECMR.2015.7324176, http://ieeexplore.ieee.org/document/7324176/
2. Kaelbling, L.P., Lozano-Pérez, T.: Hierarchical task and motion planning in the now. Proceedings - IEEE International Conference on Robotics and Automation pp. 1470–1477 (2011). https://doi.org/10.1109/ICRA.2011.5980391
3. Karaman, S., Frazzoli, E.: Sampling-based Algorithms for Optimal Motion Planning (may 2011), http://arxiv.org/abs/1105.1186
4. LaValle, S.M.: Rapidly-exploring random trees: A new tool for path planning. Tech. rep. (1998), http://msl.cs.uiuc.edu/ lavalle/papers/Lav98c.pdf
5. Li, R., Gu, D., Liu, Q., Long, Z., Hu, H.: Semantic Scene Mapping with Spatio-temporal Deep Neural Network for Robotic Applications. Cognitive Computation **10**(2), 260–271 (apr 2018). https://doi.org/10.1007/s12559-017-9526-9, http://link.springer.com/10.1007/s12559-017-9526-9

6. Martinez-Martin, E., del Pobil, A.P.: Personal Robot Assistants for Elderly Care: An Overview. pp. 77–91. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-62530-0$_5$, $http://link.springer.com/10.1007/978-3-319-62530-0_5$
7. Nico Sun, Erfu Yang, Jonathan Corney, Y.C., Ma, Z.: Semantic enhanced navigation among movable obstacles in the home environment. In: Embedded Intelligence: Enabling & Supporting RAS Technologies. pp. 68–71 (2019), https://www.ukras.org/wp-content/uploads/2019/03/UKRAS19-Proceedings-Final.pdf
8. Petr Masek, M.R.: A Task Planner for Autonomous Mobile Robot Based on Semantic Network in Advances. Advances in Intelligent Systems and Computing 393 pp. 634–639 (2016)
9. Stilman, M., Kuffner, J.J.: Navigation among movable obstacles: real-time reasoning in complex environments. Tech. rep., https://smartech.gatech.edu/bitstream/handle/1853/36417/stilman-ijhr2005.pdf
10. Strömbergsson, S., Hjalmarsson, A., Edlund, J., House, D.: Timing responses to questions in dialogue. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH pp. 2584–2588 (2013)
11. Sun, H., Meng, Z., Ang, M.H.: 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). pp. 207–212. IEEE (nov 2017). https://doi.org/10.1109/ICCIS.2017.8274775, http://ieeexplore.ieee.org/document/8274775/
12. Tenorth, M., Kunze, L., Jain, D., Beetz, M.: KNOWROB-MAP - knowledge-linked semantic object maps. In: 2010 10th IEEE-RAS International Conference on Humanoid Robots. pp. 430–435. IEEE (dec 2010). https://doi.org/10.1109/ICHR.2010.5686350, http://ieeexplore.ieee.org/document/5686350/