Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 612–621

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

An algorithm for modeling an environment and creating a semantic model of its topology

Daniel Figurowski*, Paweł Dworak

Department of Automatic Control and Robotics, West Pomeranian University of Technology, Szczecin, Poland

Abstract

This paper discusses the problem of a semantic topology map generation during the environment mapping process. A method is proposed which provides a model of a mobile robot's working space. The proposed world representation is divided into two maps: semantic topology map and metric map. The first one is created by employing graph structure which describes relations between different zones generated during exploration of the environment, and the metric one which defines zones by expandable polygonal chain of border points. The acquired hierarchical world model can be used in the path planning and decision tasks in a human-friendly way by the means of semantic sentences. Such approach increases versatility of human-robot cooperation. The proposed algorithm has been validated in an experiment and the obtained results which shows a good performance of the devised method are presented.

Keywords: environment mapping; topology graph; semantic map

1. Introduction

The design of cooperative servant robots, which would be human-friendly in usage, requires development of advanced systems and algorithms. The whole navigational system must be redefined to comply with this concept and many problems must be approached in a new way [1]. One of the key issues in development of such system is the mapping process which would gather knowledge about the environment for later use in the reasoning process. Work

* Corresponding author. Tel.: +48-91-449-53-18;
E-mail address: daniel.figurowski@zut.edu.pl

10.1016/j.procs.2021.08.063

on algorithms for building models of the mobile robots surroundings has been ongoing for many years [2, 3]. They are concerned with the construction of metric, topological and, sometimes, semantic models of space explored by a mobile robot. Their results lead to a more accurate and effective path determination and robot motion control. The use of classifiers, maps and semantic graphs will allow for a qualitative change in the means of communication with a mobile robot. It will lead to a more effective execution of tasks in an unknown or partially known working environment of the robot.

In [4] a topological map generation and navigation method are proposed to build the map of an indoor environment. A direct corridor type identification is realized by a progressive Bayesian classifier relying on a depth curve provided by a 3D sensor. Information from multiple observations is utilized to achieve a more robust performance. The derived map and Markov localization method allow the autonomous robot to localize itself and navigate freely in the indoor environment.

A knowledge-based system to provide web and cloud services for a simple robot is presented in [5]. The RoboEarth semantic mapping system is composed of an ontology to organize the concepts and relations in maps and objects, and a SLAM map providing the scene geometry and object locations with respect to the robot. Utilizing cloud services a simple robot can reliably and efficiently build the semantic maps needed to perform its everyday tasks. The system enables semantic map building for a novel environment while exploiting available prior information about the environment, or alternatively, searches for a novel object in a previously known environment with an increased efficiency due to the reasoning utilizing a semantically annotated map.

A semantic map representation called Multiversal Semantic Map presented in [6] keeps the sets of different groundings, or universes, as instances of ontologies annotated with the obtained beliefs for their posterior exploitation. A multi-hierarchical semantic map supplements a metric map of the explored environment. Conditional Random Fields help in dealing with the uncertainty inherent to the grounding process and in solving contextual ambiguities and relations among concepts.

The method presented in [7] allows an on-line unsupervised learning. A self organizing mapping (SOM) module incrementally creates a topological map of the environment enriched with objects recognized around each topological node. A semantic map in [8] is a part of three layer model containing also a working memory and episodic memory layers. Each comprises a self-organizing adaptive recurrent incremental network whose task is to cluster sensory information and to learn relationships between them.

The speech signals given by a user are recognized and allow addition of semantic information to an environment model of a self-localizing robot in [9]. The robot learns the spatial concepts by integrating multi-sensor data and generates a semantic map (SpCoMapping). A nonparametric Bayesian algorithm automatically estimates an adequate number of categories.

A semantic description of basic indoor structures (corridors, walls, open areas) is added to OpenStreetMap (OSM) — the topological and geometrical description of the robot environment in [10]. Such graph-based map representation allows to exploit the hierarchical structure of the graphs and utilize specific area model in a topological path planning.

Semantic maps of the environment are constructed in [11] with the use of the end to end probabilistic deep networks (TopoNets). The learning algorithm uses partial sensory observations and noisy topological relations discovered by a robot exploring large-scale office spaces. It allows to create a deep network which is able to work in real-time making tractable and exact inference and thus enabling a mobile robot to spatially understand knowledge at different abstraction scale.

The long-term autonomy of the system and generation of human-understandable models of the environment still remains a challenge and permeates most works devoted to the problem of environmental mapping. The aforementioned models would allow for transparent human-robot cooperation. Devised solutions should enable environment exploration and on-line modification of metric, topological and semantic maps while minimizing the number of necessary parameters used for world description thus leading to a decrease in memory consumption and faster data processing.

The following paper focuses on generation, in the process of environment mapping, of a compact world model with a semantic representation of topology. As a solution to this problem an algorithm is proposed which allows for the generation and modification of environmental maps using polygons and graph theory while assuming discrete sensor motion. The polygon tools were used in the process of building a metric map of the mobile robot's environment which allows for efficient manipulation of the map's data based on the semantic description of the analyzed space. As a result, the created metric representation is consistent and semantically defines the areas of the environment. At the same time, it allows for the creation of a semantic topology of the robot's surroundings in the form of a directed graph describing the relations between the polygon zones which enables the use of this data in fast path planning algorithms.

The article is organized as follows. In Section II a description of the problem is given as well as the details of the proposed method of environment mapping. Section III lists the simulation assumptions and presents research results. Finally, conclusions and a plan for further work are presented in Section IV.

2. Method description

A mobile robot located in a two-dimensional Cartesian representation of the world M_w has the task of building a semantic topology graph M_s during the process of mapping its surroundings. The robot is moving along a path P_d creating its own metrical world representation M_m. Proposed solution for this problem is shown in Figure 1 and its working is described in subsequent algorithms.

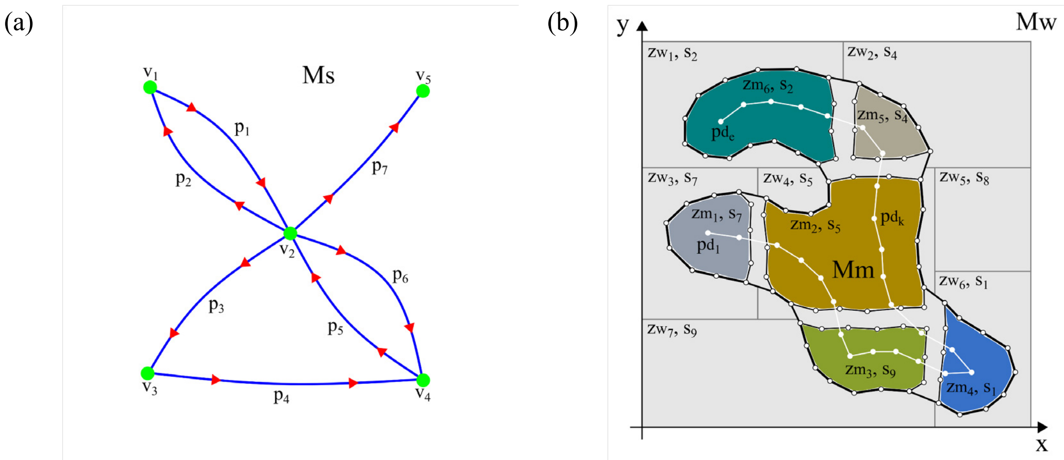


Fig. 1. The idea of (a) semantic topology graph generation during (b) environment mapping.

The world map definition used in this paper is assumed as follows: map $M_w = \{z_{w_1}, z_{w_2}, \dots, z_{w_n}\}$ is a set of n semantic zones z_w. Each zone is a tuple $z_w = (R_z, s_w)$ which contains a semantic descriptor s_w from the set $S_w = \{s_{w_1}, s_{w_2}, \dots, s_{w_g}\}$ and a tuple $R_z = (p_l, p_h)$, such that $p_l, p_h \in \mathbb{R}^2$ describing a rectangular area. The semantic set S_w contains unique g types of space. Rectangular shape of the world is a direct result of developing the environment as a vector graphic, but such description simplifies search algorithms and saves memory.

Algorithm 1: Main program

Require: World map M_w, Movement path P_d, Sensor parameters U_s;

$M_m(0) \leftarrow \emptyset, M_s(0) \leftarrow \emptyset$
for $k = 1, \dots, |P_d|$ **do**
  $p_r \leftarrow P_d(k)$
  $M_u \leftarrow$ ExtractSensorMap(M_w, U_s, p_r)
  $[M_m(k), M_s(k)] \leftarrow$ UpdateMemoryMap$(M_m(k-1), M_s(k-1), M_u, p_r)$
end

Result: $[M_m, M_s]$

The main program loop of the proposed method is presented in Algorithm 1. During initialization process parameters of the simulation environment are prepared i.e. world map M_w, movement path P_d and sensor parameters U_s. The movement path (acquired from an external generator) is an ordered set $P_d = \left(p_{d_1}, p_{d_2}, \dots, p_{d_k}, \dots, p_{d_e}\right), p_{d_k} \in \mathbb{R}^2$, where $k = 1,2,3,\dots,e$ is a discrete cycle index (time) of the proposed method and e is the number of path points. The sensor parameters is a tuple $U_s = \left(u_{s_D}, u_{s_R}, u_{s_Q}\right)$ which comprises of sensor scanning directions quantity $u_{s_D} \in \mathbb{N}^+$, sensor scanning range $u_{s_R} \in \mathbb{R}_{>0}$, and sensor scanning points quantity $u_{s_Q} \in \mathbb{N}^+$. Robot's representation of the working space in form of a metric map M_m and a semantic topology graph M_s are cleared from the memory before starting the process of environment mapping. After initialization process the algorithm cyclically repeats robot's movement, sensory action and memory processing until the path end p_{d_e} is reached. During the movement state current robot's location p_r is updated to the next path point $P_d(k)$ according to the algorithm's present iteration k. After the robot executes its movement the function extracting the sensor map M_u from the measurements of world map M_w is executed. Finally, the acquired data is used to update the robot's knowledge about the mapped environment.

Algorithm 2: Extract sensor map

Require: World map M_w, Sensor parameters U_s, Current robot location p_r;

$M_u \leftarrow \emptyset, \left\lfloor u_{s_D}, u_{s_R}, u_{s_Q} \right\rfloor \in U_s$
for $i = 1, \dots, u_{s_D}$ **do**
  $\alpha \leftarrow 2\pi\,(i-1)/u_{s_D}$
  for $j = 1, \dots, u_{s_Q}$ **do**
    $p_{xy} \leftarrow p_r + j\, u_{s_R}/u_{s_Q} * [\sin\alpha \quad \cos\alpha]$
    $U_{xy}(j) \leftarrow \mathrm{ReadMap}\left(M_w, p_{xy}\right)$
  end
  $S_{xy} \leftarrow \emptyset, P_{xy} \leftarrow \emptyset, O_{xy} \leftarrow \emptyset$
  if ¬IsObstacle$\left(U_{xy}(1)\right)$ **then**
    for $j = 2, \dots, u_{s_Q}$ **do**
      if IsObstacle$\left(U_{xy}(j)\right)$ **then**
        $S_{xy} \leftarrow S_{xy} \cup \mathrm{GetSemantics}\left(U_{xy}(j-1)\right)$
        $P_{xy} \leftarrow P_{xy} \cup \mathrm{GetPoint}\left(U_{xy}(j-1)\right)$
        $O_{xy} \leftarrow O_{xy} \cup \{\text{"obstacle"}\}$
        break
      else if IsPassage$\left(U_{xy}(j)\right) \neq$ IsPassage$\left(U_{xy}(j-1)\right)$ **then**
        if IsPassage$\left(U_{xy}(j)\right)$ **then**
          $S_{xy} \leftarrow S_{xy} \cup \mathrm{GetSemantics}\left(U_{xy}(j-1)\right)$
          $P_{xy} \leftarrow P_{xy} \cup \mathrm{GetPoint}\left(U_{xy}(j-1)\right)$
        else
          $S_{xy} \leftarrow S_{xy} \cup \mathrm{GetSemantics}\left(U_{xy}(j)\right)$
          $P_{xy} \leftarrow P_{xy} \cup \mathrm{GetPoint}\left(U_{xy}(j)\right)$
        end
        $O_{xy} \leftarrow O_{xy} \cup \{\text{"passage"}\}$
      else if $j = u_{s_Q}$ **then**
        if ¬IsPassage$\left(U_{xy}(j)\right)$ **then**
          $S_{xy} \leftarrow S_{xy} \cup \mathrm{GetSemantics}\left(U_{xy}(j)\right)$
          $P_{xy} \leftarrow P_{xy} \cup \mathrm{GetPoint}\left(U_{xy}(j)\right)$
          $O_{xy} \leftarrow O_{xy} \cup \{\text{"free"}\}$

```
30 |       |       |       end
31 |       |       end
32 |       end
33 |   end
34 |   U_r(i) ← (S_xy, P_xy, O_xy)
35 end
36 S_u ← GetUniqueSemantics(U_r)
37 foreach s_u ∈ S_u do
38 |   P_u ← ExtractPoints(s_u, U_r)
39 |   O_u ← ExtractOccupancy(s_u, U_r)
40 |   z_u ← CreateZone(s_u, P_u, O_u)
41 |   M_u ← M_u ∪ z_u
42 end
Result: M_u
```

The sensory action presented in Algorithm 2 is divided into three parts: measurement, data analysis and map extraction. During the measurement phase virtual sensor acquires ordered set of data $U_{xy} = (u_1, u_2, \ldots, u_n)$, where n is the number of scanned points u_{s_Q}. Each element of this set is a tuple $u_{xy} = (s_{xy}, p_{xy})$ which contains spatial point information of semantic description $s_{xy} \in S_w$ in measurement location $p_{xy} \in \mathbb{R}^2$. Such sets are generated for a number u_{s_D} of evenly angular-spaced scanning actions. The acquired points from a single scan action are colinear and equally-spaced with regards to the current robot location p_r and sensor range u_{s_R}.

The data analysis stage is responsible for obtaining a characteristic data set U_r from an ordered set U_{xy} by application of validation rules for consecutive points or neighboring points pairs. The hierarchically processed rules look for passages (Alg. 2, line 16) by analyzing semantics of the ordered set U_{xy} from the beginning to the first occurrence of an obstacle (Alg. 2, line 11) or last point. The third rule (Alg. 2, line 25) defines the free space of the last point as a consequence of sensor range limitation. This leads to memorization of such point when earlier rules do not apply. The resulting set is a tuple $U_r = (S_{xy}, P_{xy}, O_{xy})$ which consists of three ordered sets containing semantic descriptions S_{xy}, measurement locations P_{xy} and occupancy definitions O_{xy} of selected characteristic points.

The last part of the algorithm extracts unique semantic descriptions S_u from the set U_r and creates a sensor map M_u, which is constructed from polygonal zones z_u. Each zone z_u is generated from the ordered subsets of measurement locations P_u and occupancy definitions O_u based on the grouping of data by their semantic description $s_u \in S_u$. The creation of the map zone is achieved by the usage of a convex hull generator function and a custom geometry computation to create concave shapes.

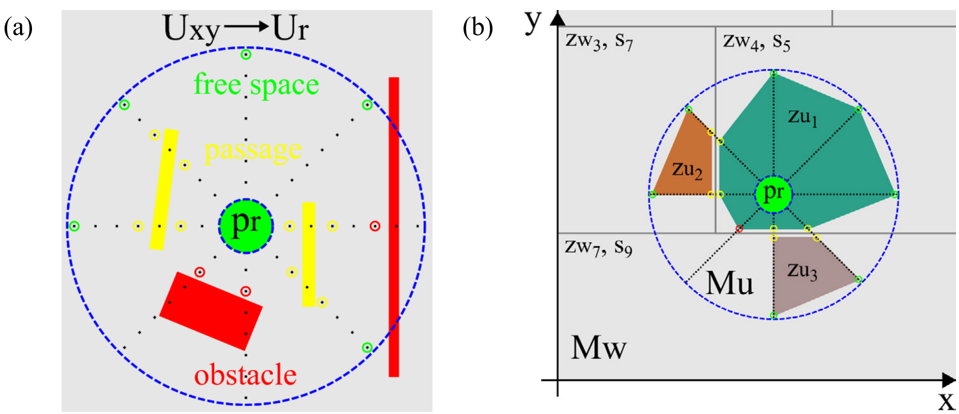


Fig. 2. Sensory action, (a) measurement and data analysis, (b) map extraction.

The concept of a sensory action is presented in Figure 2. The left image shows the measurement stage (the set U_{xy} gathered in current robot's location p_r is represented by black dots) and generation of the set U_r (found obstacles – red, free space – green and passages – yellow circles). The right image visualizes creation of the sensor map M_u from polygonal zones z_u.

Algorithm 3: Update memory map

Require: Metric map M_m, Semantic map M_s, Sensor map M_u, Current robot location p_r;

$z_r \leftarrow$ GetNearestZone(M_u, p_r)
$s_r \leftarrow$ GetSemantics(z_r)
$S_m \leftarrow$ GetSemantics(M_m)
if $S_m \ni s_r$ **then**
    $z_m \leftarrow$ GetZone(s_r, M_m)
    $M_m \leftarrow M_m \setminus z_m$
    $M_m \leftarrow M_m \cup (z_m \oplus z_r)$
else
    $M_m \leftarrow M_m \cup z_r$
    $V(M_s) \leftarrow V(M_s) \cup V(s_r)$
end
$M_u \leftarrow M_u \setminus z_r$
foreach $z_u \in M_u$ **do**
    $s_u \leftarrow$ GetSemantics(z_u)
    if $S_m \ni s_u$ **then**
        $z_m \leftarrow$ GetZone(s_u, M_m)
        $M_m \leftarrow M_m \setminus z_m$
        $M_m \leftarrow M_m \cup (z_m \oplus z_u)$
        if $E(M_s) \not\ni E(s_r, s_u)$ **then**
            $p_u \leftarrow$ GetNearestPoint(z_u, p_r)
            $E(M_s) \leftarrow E(M_s) \cup E(s_r, s_u, p_u)$
        end
    else
        $M_m \leftarrow M_m \cup z_u$
        $p_u \leftarrow$ GetNearestPoint(z_u, p_r)
        $V(M_s) \leftarrow V(M_s) \cup V(s_u)$
        $E(M_s) \leftarrow E(M_s) \cup E(s_r, s_u, p_u)$
    end
end

Result: $[M_m, M_s]$

The memory processing presented in Algorithm 3 starts with procedure which acquires the robot's semantic position s_r according to the robot's current location p_r and newly acquired sensor map M_u based on the nearest sensory zone z_r. This information is needed during the task of updating semantic topology graph edges which is realized concurrently with the update of metric map M_m. Firstly, existence of this semantic position s_r is validated according to the current semantic memory state S_m. If it is not existent, it will be added to the vertices of semantic topology graph M_s, and the polygonal zone z_r will be merged with metric map M_m. If it exist, then such zone z_m will be extracted from the metric map M_m with accordance to semantic equality and will be fused with a new zone

z_r acquired by the robot's sensor. Update of the semantic topology graph M_s is skipped because in such situation vertices must have been created earlier during the first occurrence of zone with the same semantic description.

After the first processing stage the rest of zones from the sensor map M_u are processed similarly. The only difference lies in the graph update process which now generates edges in respect to robot's semantic current location s_r. New zones z_u that do not exist semantically are added to the metric map M_m and vertices for them are created in the semantic topology graph M_s as well as appropriate directed edges. Such edge describe the source as the robot's semantic current location s_r, destination as semantics s_u of a new zone z_u and a nearest point of transition to a new zone z_r from robot's perspective. Already existing semantic zones are fused together and a new edge is created only when it was not defined earlier.

3. Results

To verify correctness and efficiency of the proposed algorithm the following tests were caried out with the assumed scenario:

- A simulated mobile robot located at the starting point (1, 6) moves along the path points in a 2D space, which was set up to represent a corridor loop with four neighboring rooms. Selected path points were disturbed with random shifts (± 0.1 m) from their original locations;
- The data gathered from a virtual sensor is always matching the real semantics of the environment, measured data is binary characterized. The range of the sensor was set to 2 meters with a quantization of 0.1 meters;
- The simulation of the proposed algorithm is executed in discrete time manner where time is defined as consecutive path points indices. In each of these points robot executes one cycle of the algorithm and updates its knowledge about the environment. The simulation was finished when the robot reached its path's end point;

The research was carried out in Matlab 2019b using a computer with the following specification: CPU – AMD Ryzen Threadripper 1920X 3.6 Ghz, RAM – 4x16 GB DDR4-3200 CL14, GPU – Nvidia GeForce GTX 750Ti 2 GB, HD – Samsung SSD 970 PRO 512GB.

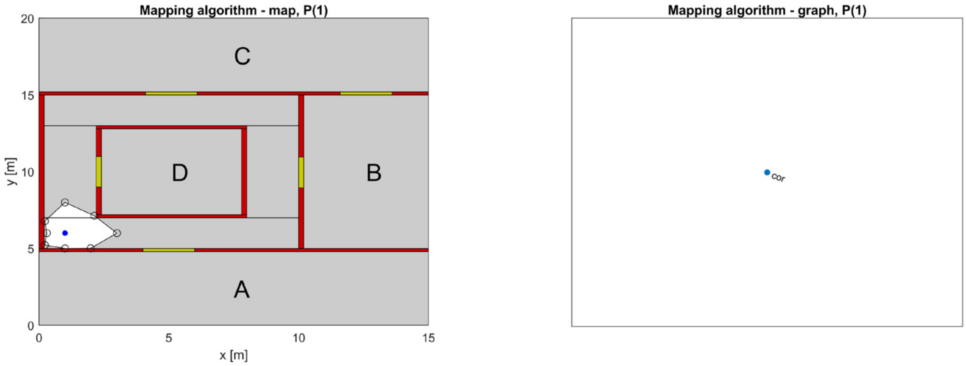
Table 1 shows the results of the simulation experiment. It contains time statistical values of the mapping algorithm. As it may be observed the sampling point calculation times stay in some bounds and an increase in calculation time was not observed. Acquired world model is compact and has low memory demanding, allowing to longer exploration of large spaces and faster computation in planning processes.

Table 1. Results of simulation tests.

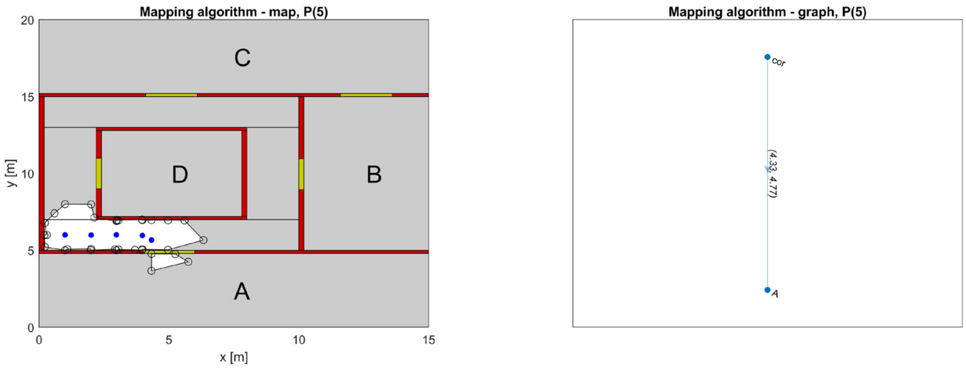
Statistic function	Algorithm time (s)	Sensor time (s)			Map update time (s)	Graph update time (s)
		Measurement	Analysis	Extraction		
Min	0.0629	0.0257	0.0278	0.0016	0.0018	0.0009
Max	0.0829	0.0335	0.0410	0.0037	0.0106	0.0026
Mean	0.0728	0.0290	0.0338	0.0025	0.0059	0.0014
Std	0.0040	0.0010	0.0032	0.0006	0.0018	0.0006

The resulting evolution of the environment model are shown in Figure 3a-3g as a sequence of metrical maps and semantic topology graphs. Selection of time points was based on the change in the graph's structure. Map images present metrical environment representation where an unexplored space is dimmed and the explored one is defined by the border points chains. The zones with labels represent different types of rooms which are connected to a corridor. Such connections are presented as passages with yellow color. Impassable spaces, e.g. walls, are marked by red color. The graph images present topology of the explored environment where nodes are specific locations and directional connections define passage point between nodes.

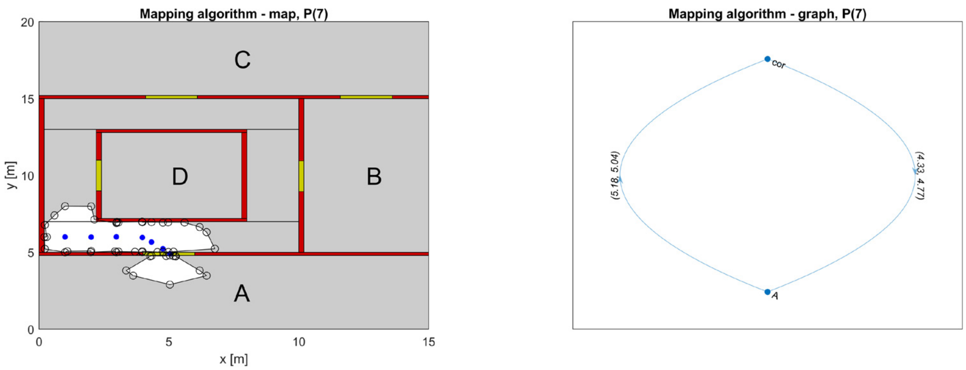
(a)


Mapping algorithm - map, P(1)
Mapping algorithm - graph, P(1)
C
D
B
A
y [m]
x [m]
cor

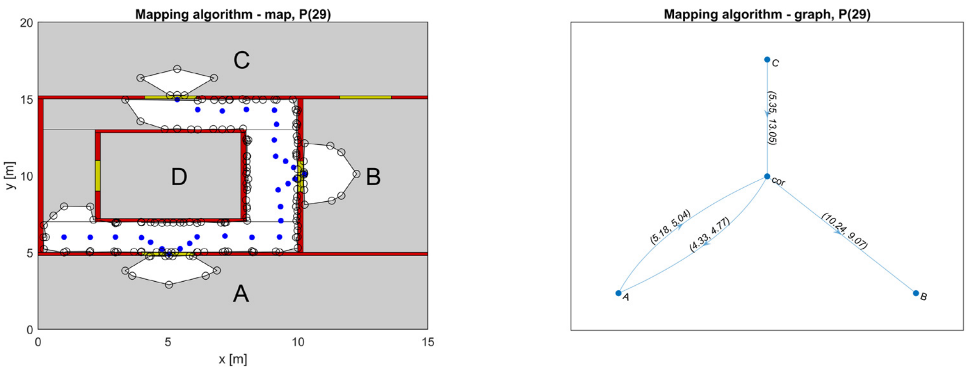
(b)


Mapping algorithm - map, P(5)
Mapping algorithm - graph, P(5)
C
D
B
A
y [m]
x [m]
cor
(4.33, 4.77)
A

(c)


Mapping algorithm - map, P(7)
Mapping algorithm - graph, P(7)
C
D
B
A
y [m]
x [m]
cor
(5.18, 5.04)
(4.33, 4.77)
A

(d)


Mapping algorithm - map, P(29)
Mapping algorithm - graph, P(29)
C
D
B
A
y [m]
x [m]
C
(5.35, 13.05)
cor
(5.18, 5.04)
(4.33, 4.77)
(10.24, 9.07)
A
B

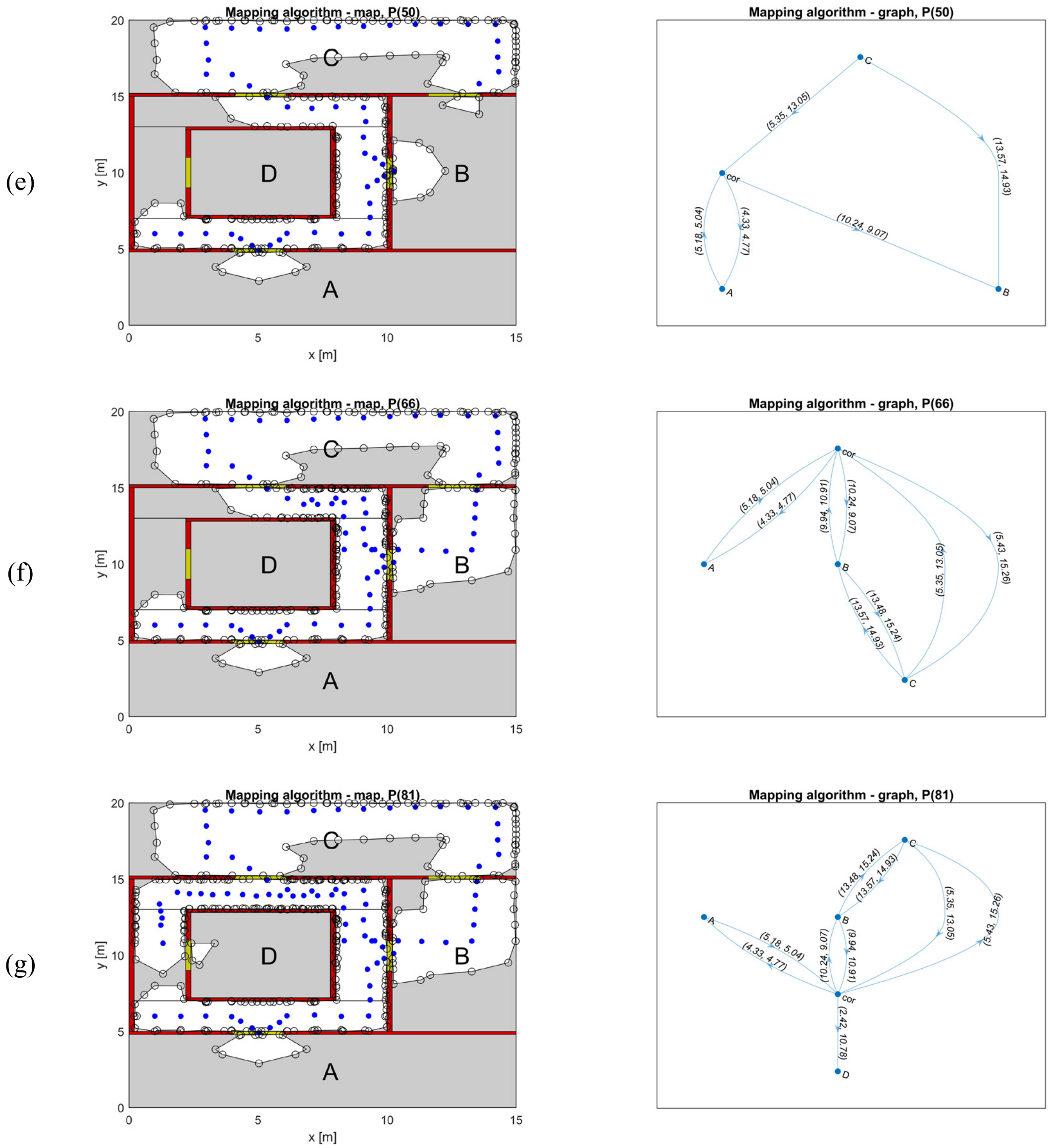


Fig. 3. Environment maps generated by the proposed algorithm. Left column – metrical map. Right column – semantic topology graph.

In step one (Fig. 3a) a metrical map was generated directly from the sensor measurements in this initial position. First node of a semantic topology graph for the “corridor” zone was created. In the following steps the mapping process expands polygonal border chain of “corridor” zone in the metrical map. In the fifth step (Fig. 3b) robot’s sensor detects new zone “A” which leads to creation of a new node in the sematic graph. It is also responsible for the creation of the first directed edge from “corridor” to zone “A”. In addition, information about the nearest passage point - from robot’s perspective to a neighboring zone - is memorized in newly created edge. At step seven (Fig. 3c), the robot’s position is closer to zone “A”. This leads to an update of the semantic graph and creation of new directed edges from zone “A” to “corridor”.

This process is repeated in next robot’s movements but sometimes the resulting direction of edges can be counterintuitive, e.g. at step twenty-nine (Fig. 3d) the robot is placed in the zone “C”. However, the edge that would allow transition from “corridor” to “C” is not present. This behavior is a consequence of sensory data acquisition and a very peculiar robot placement in which the acquired data was insufficient to create a distinguishable zone to allow detection of such event and update a semantic graph accordingly. At step sixty-six (Fig. 3f) it can be seen that this misdetection can be corrected when the robot moves for the second time near the zone “C” passage. The final result of the proposed method at step eighty-one (Fig. 3g) presents a metrical map and a semantic topology graph after passing the whole movement path.

4. Conclusions

The simulation results prove the validity of the proposed method of environment mapping with semantic topology representation using polygonal framework and semantic information. The presented algorithm was tested experimentally for spatial description of metric map and topology graph of environment and was proven to be efficient. The obtained results show consistence in the processing time of the algorithm. Resulting metric model of the explored space is sufficient to allow a mobile robot to safely operate in an environment.

In the future work it is planned to add data reduction process that would allow for further elimination of excessive number of points in metric map border chains. Finally, the algorithm will be adopted and integrated into an autonomous exploration method.

References

[1] Crespo, J., Castillo, J. C., Mozos, O. M., Barber, R.: Semantic Information for Robot Navigation: A Survey. Appl. Sci. 10(2), 497 (2020), doi: 10.3390/app10020497.

[2] Cadena, C. et al.: Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. IEEE Trans. Robot. 32(6), 1309–1332 (2016), doi: 10.1109/TRO.2016.2624754.

[3] Kostavelis, I., Gasteratos, A.: Semantic mapping for mobile robotics tasks: A survey. Robot. Auton. Syst. 66, 86–103 (2015), doi: 10.1016/j.robot.2014.12.006.

[4] Cheng, H., Chen, H., Liu, Y.: Topological Indoor Localization and Navigation for Autonomous Mobile Robot. IEEE Trans. Autom. Sci. Eng. 12(2), 729–738 (2015), doi: 10.1109/TASE.2014.2351814.

[5] Riazuelo, L. et al.: RoboEarth Semantic Mapping: A Cloud Enabled Knowledge-Based Approach. IEEE Trans. Autom. Sci. Eng. 12(2), 432–443 (2015), doi: 10.1109/TASE.2014.2377791.

[6] Ruiz-Sarmiento, J.-R., Galindo, C., Gonzalez-Jimenez, J.: Building Multiversal Semantic Maps for Mobile Robot Operation. Knowl.-Based Syst. 119, 257–272 (2017), doi: 10.1016/j.knosys.2016.12.016.

[7] Sousa, Y. C. N., Bassani, H. F.: Incremental Semantic Mapping with Unsupervised On-line Learning. In 2018 International Joint Conference on Neural Networks, 1–8 (2018), doi: 10.1109/IJCNN.2018.8489430.

[8] Chin, W. H., Kubota, N., Ju, Z., Liu, H.: Navigate to Remember: A Declarative Memory Model for Incremental Semantic Mapping. In Intelligent Robotics and Applications, 142–153 (2019), doi: 10.1007/978-3-030-27538-9_13.

[9] Katsumata, Y., Taniguchi, A., Hagiwara, Y., Taniguchi, T.: Semantic Mapping Based on Spatial Concepts for Grounding Words Related to Places in Daily Environments. Front. Robot. AI, 6 (2019), doi: 10.3389/frobt.2019.00031.

[10] Naik, L., Blumenthal, S., Huebel, N., Bruyninckx, H., Prassler, E.: Semantic mapping extension for OpenStreetMap applied to indoor robot navigation. In 2019 International Conference on Robotics and Automation (ICRA), 3839–3845 (2019), doi: 10.1109/ICRA.2019.8793641.

[11] Zheng, K., Pronobis, A.: From Pixels to Buildings: End-to-end Probabilistic Deep Networks for Large-scale Semantic Mapping. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3511–3518 (2019), doi: 10.1109/IROS40897.2019.8967568.