Topological Robot Localization based on Visual Object Detection and Node Classification for indoor navigation

Sergio Lafuente-Arroyo[1][0000-0002-4841-2501], Saturnino Maldonado-Bascón[1][0000-0001-6472-5359], Diego Delgado-Mena[1][0000-0002-5672-1865], Juan Antonio Castro-García[2][0000-0003-1128-6879], and Roberto Javier López-Sastre[1][0000-0002-2477-0152]

[1] Universidad de Alcalá, Alcalá de Henares (Madrid), España
GRAM, Departamento de Teoría de la Señal y Comunicaciones,
{sergio.lafuente,saturnino.maldonado,diego.delgadom,robertoj.lopez}@uah.es
https://gram.web.uah.es

[2] Departamento de Tecnología Electrónica, E.T.S.I. Informática. Universidad de Sevilla, España.
jacastro@us.es

Abstract. Topological maps allow to work with simplified diagrams as an abstraction of the environment. These maps lack scale but the relationship between points is maintained. In this work we propose a novel method for robot localization based on the search for the structure of the environment that the robot has visited during the navigation. From the information extracted by the sensors on board the mobile platform, our approach is able to extract information corresponding to the graph structure of the visited area, such as the types of nodes, the detected objects and their relative locations. In our proposal an evaluation function provides the probabilities that a detected path corresponds to the different edges of the topological map. Experiments are conducted in a real scenario where the topological map is available. Quantitative evaluations demonstrate that the system is able to locate the robot even with a non-accurate detection of objects.

Keywords: Assistive Robot · Topological Map · Node Classification · Object detection · Robot localization.

1 Introduction

Robot navigation within large-scale, semi-structured environments deals with various challenges such as location about traversable spaces. This makes it harder to deal with uncertainties that are present in the context of real-time robotics applications. A topological graph, as defined by Simhon and Dudek [1], can be of a great benefit in many robotic navigation-related tasks. This implies an abstraction of the environment in which the important elements are defined along with the transitions among them. In complex real indoor environments, such as hospitals, elderly care facilities and office buildings, the structure presents high level of symmetry and usually consists of many corridors in which rooms are distributed on both sides.

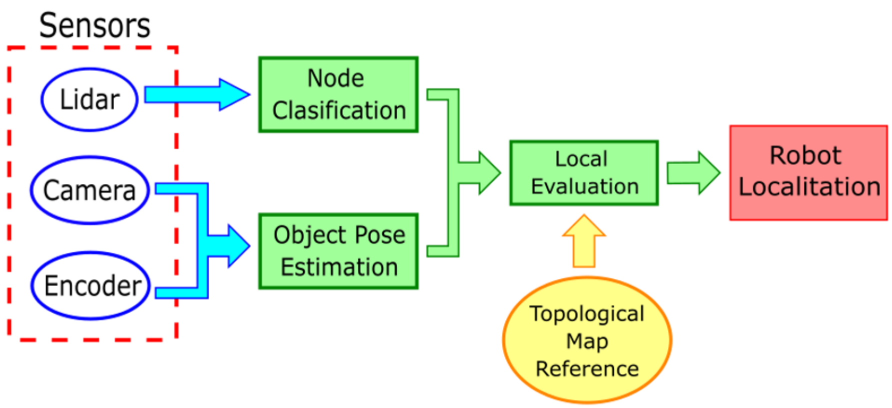
In this paper, we propose to build a topological representation in which key entities correspond to detected objects and locations with relevant changes of trajectory. More specifically, we introduce topological nodes for relevant changes of direction, such as an end of aisle or a bifurcation, where several outlets are possible. Each entity accessible from another entity is connected using an edge. However, robot localization require more detailed information about the presence of objects and their relative locations. Thus, objects are considered as references in the topological map and we will refer to them as sub-nodes. According to this structure, a route is specified by an edge or sequence of edges, where each edge is defined by the classes of its terminal nodes and the relative positions of certain objects with respect to a reference node. It is worth noting that in complex, highly symmetric environments, we can find edges with a very similar structure. Only the class of one of its terminal nodes or the relative positions of detected objects can be crucial to differentiate similar edges.

In order to give high-level abstraction instructions to the robot, representations of the environment similar to the human mode of interpretation are needed. In this context, it is not necessary to handle very precise metric information and may not even be considered. More specifically, we propose a topological representation using relative distances between objects of a complex scenario.

Figure 1 shows a block diagram of the proposed approach. The complete process is performed automatically using the topological map of the environment as input and consists of three main modules: 1) node classification based on depth information, where nodes represent locations that involves a mandatory or optional change of the robot trajectory, 2) estimation of relative position of objects in the topological map from visual detection and tracking, and 3) local evaluation in order to estimate robot localization in the topological map based on nodes and sub-nodes information from the outputs of the two previous modules.

In summary, the main contributions of this work are as follows:

– We introduce a evaluation function to locate the robot. It allows to evaluate the degree the similarity between the path followed by the robot and the possible routes of the topological map. The function takes as inputs the information extracted from nodes and sub-nodes in the path followed by the robot and the topological reference map.
– We use the semantic information from the objects found by means of a trained YOLO-v3 [2] based detector to automatically detect sub-nodes of the topological map.
– Although numerous tracking algorithms can be found in the literature [3, 4], we introduce a novel approach adapted to our approach. We define a tracking vector that takes into account both the evolution of coordinates of the objects in the image plane and the robot movements from encoders.



(a)

Fig. 1. Overview of the proposed system.

In order to establish the associations of objects, we have implemented an efficient Support Vector Machine (SVM)[5]-based approach.

This paper is organized as follows. We start by an overview of related work in Section 2. In Section 3, we describe the details of the modules of the complete system. Results are presented in Section 4. Finally, Section 5 concludes the paper and outlines future research.

2 Related Work

Navigation systems are based on metric, topological, and semantic maps, depending on the level of abstraction of the environment representation [6]. In the literature, a large variety of solutions to this problem is available. One intuitive way of formulating SLAM is to use a graph whose nodes correspond to the poses of the robot at different points in time and whose edges represent constraints between the poses. Regarding topological navigation, since the first developments, the global conception of the system has attracted the interest of several authors. Surveys of models for indoor navigation are provided by [7–9] among others. Comparison of various graph-based models is provided in a very clear way in Kielar et al. in [10].

In [11], a topological map based algorithm is described to explore and construct the map of a unknown indoor environment. The features to close loop are not described but authors state that is possible to localize the robot based on them. In this case, eight nodes types are considered according to the outlets in each node. Although panoramic representation is considered in some works [12], we use standard 640x480 images to obtain features for localization and navigation.

Devendra et al. in [13] described a image-goal navigation solution with a different proposal to ours. In fact, they avoid metric information to reduce the challenge of precise representation. In this work they combine exploration without an objective and navigating with a goal.

Inspired by the work [16] we detect nodes for the topological map and to classify them into different categories regarding to the LIDAR signature.

3 Topological localization

The proposed topological localization method adopts a three-stage framework based on the sensors on board the robot. The sensing part is composed of an Intel RealSense D435 camera and a LIDAR. First, a node classification is performed on the LIDAR signature, which provides depth information of the scenario in which the robot is at any given moment. At the second stage, objects of interest are detected and tracked to extract their relative positions. Finally, the third state estimates the probabilities of occurrence of the possible paths of the network according to the information generated in the previous states. In the following subsections, we will describe the implementation of the above three modules.

3.1 Node classification

The first stage of the whole system consists of the extraction of nodes in indoor environments. A previous work [16] is used for this task. Here nodes represent relevant changes of direction, such as an end of aisle or a bifurcation, where several outlets are possible. As depth information for node classification, we rely on LIDAR scanning. Regarding to this criterion, common corridor structures can be classified into four node categories:

- End node: there is no outlet at the front, neither from the left nor from the right of the corridor.
- Node 'T': there are two outlets for the agent since the corridor presents two lateral bifurcations.
- Node 'L': the path presents an marked change of direction.
- Cross node: the corridor presents three or more outlets.

The LIDAR signature feeds the input of our classifier whose output determines the node type. Because the efficiency of Support Vector Machines (SVMs), we used this technique as base of our classification module. Figure 2 shows two examples, where panoramic views and LIDAR signatures are represented. It is important to point out that panoramic images are not used in the system and are only included in the figure as a representation of the environment for a better understanding. The green and red boxes show explorable (free space) and non-explorable directions, respectively.

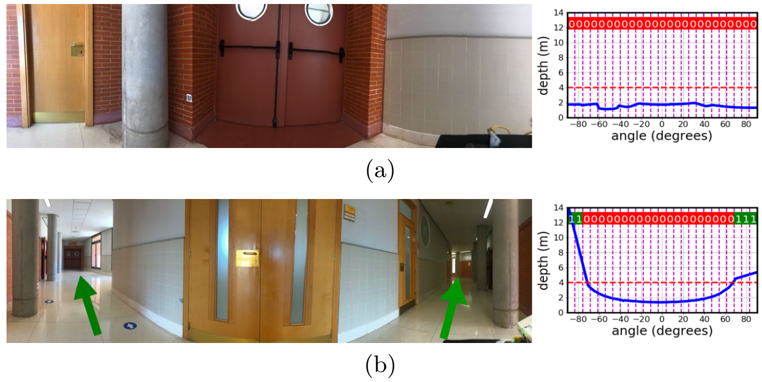


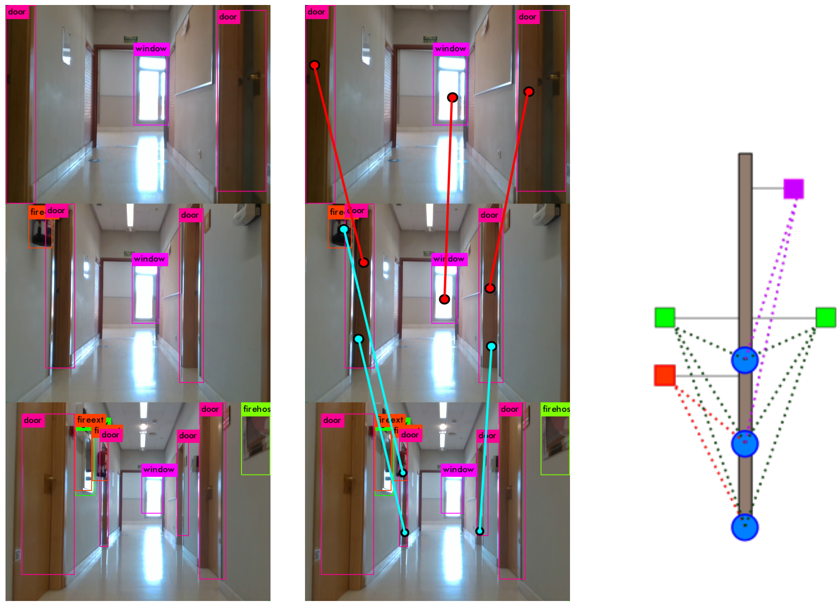
Fig. 2. Examples of node signatures. (a) End node. (b) Node T.

3.2 Object Pose Estimation

Object pose estimation involves a previous visual detection and tracking of objects of interest. A YOLO network has been trained to detect the most common objects in semi-structured indoor environments. In this work, we focused on the following nine categories: window, door, elevator, fire extinguisher, plant, bench, light box, firehose and column. Figure 3 summarises part of the process. In detail, Fig. 3(a) represents the outputs of our YOLO network for three consecutive images (from down to top) captured by the camera mounted on the mobile robot during its movement (each detection is marked with its label and bounding box), whereas Fig. 3(b) indicates the association obtained for three objects (a window, two doors and a fire extinguisher) with the proposed tracking algorithm. As result of the object pose estimation, Fig.3(c) illustrates the scheme generated. In this last figure, blue points represent consecutive positions of the robot in the environment and squares indicates different categories of objects (fire extinguisher is shown in red, doors in green and window in purple). The grey line represents the straight movement and the perpendicular black line is the position of the objects in the topological representation.

The purpose of tracking algorithm is to establish possible correspondences between objects of each pair of consecutive captures: frames $i-th$ and $(i+1)-th$. It is worth noting that tracking is applied independently for each category of objects. We propose a novel tracking vector which considers both the object information in the image and the odometry information from the robot encoders in its motion. Thus, our tracking vector $\mathbf{r} = (r_1, r_2, \ldots, r_8)^T$ encompasses the following eight attributes:

- r_1: distance travelled by the robot in cm between two consecutive captures.
- r_2: angle rotated by the robot in grades. Since a significant change of direction involves the detection of a new L-node, the movement between two nodes should be approximately straight. This is the reason why we consider



(a) (b) (c)

Fig. 3. Generation of the scheme for position estimation. (a) Consecutive input images with detected objects from down to top. (b) Associated objects by the tracking algorithm. (c) Resulting scheme for position estimation.

that the movement in an edge between two nodes should be approximately straight and we only take into account those movements with angle lower than a given threshold equal to 10 degrees.

- r_3: coordinate x_i of the centroid of the detected object by YOLO in the $i-th$ frame in pixels.
- r_4: coordinate y_i of the centroid of the detected object by YOLO in the $i-th$ frame in pixels.
- r_5: difference Δx_i of the x_i centroid object between $i-th$ frame and $(i+1)-th$ frame in pixels.
- r_6: difference Δy_i of the y_i centroid object between $i-th$ frame and $(i+1)-th$ frame in pixels.
- r_7: increase Δx_{i-1} of the previous displacement in pixels.
- r_8: increase Δy_{i-1} of the previous displacement in pixels.

The last two attributes r_7 and r_8 are denoted by the vector of displacement $\mathbf{d} = (\Delta x_{i-1}, \Delta y_{i-1})^T$, which represents the (x, y) movement in (i-2)-th and (i-1)-th frames in the case of having a previous association. As a particular case, the vector of displacement is set to $(0, 0)^T$ in case of initialisation of the tracking object.

In order to determine the possible correspondences, the system computes the tracking vector for all pairings of the same class objects. A trained SVM provides as outputs the correspondences between objects. With the aim of get similar ranges for each vector component, the tracking vector is normalized.

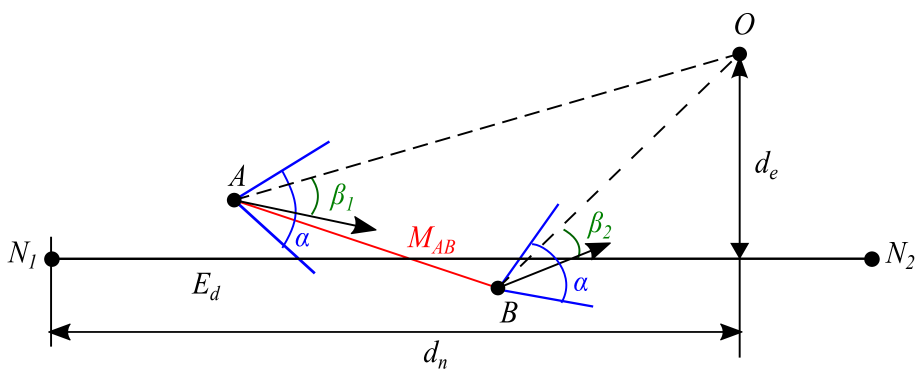


Fig. 4. Object Pose estimation

The object position in the topological map is given as a relative distance with respect to a reference node. Figure 4 shows an scheme for object pose estimation in the topological map where N_1 and N_2 represent the terminal nodes of the edge E_d, being N_1 the reference node. Here A and B denote the poses of the robot camera in the movement M_{AB}. In the two consecutive captured frames, the detected object O is projected in the plane image with angles β_1 and β_2, respectively. The parameter α denotes the camera aperture, whereas d_e and d_n represent, respectively, the distances from the object to the edge and to the reference node. In this work we only estimate the distance d_n to determine the order of appearance of the objects in the edge.

3.3 Local Evaluation

The purpose of this stage is to provide an automatic location of the robot based on the node and sub-nodes information extracted at the outputs from previous modules. Our goal is to find an efficient technique for computing the probabilities of that the path travelled by the robot corresponds with each one of the possible paths in the topological map, which is used as input. Thus, we propose a evaluation function that provides the weights in each iteration and the highest value weight determines the path that best fits the structure detected on the route.

In this approach we consider the movement from a first to a second node, where both nodes may not be adjacent as in the case of a route that passes through P nodes. Taking into account node and sub-nodes information, which provides node types and relative pose estimation objects, we propose a weight associated to the probability that the current movement corresponds in the reference topological map to the path that connect nodes α_1 and α_P with the sequence of nodes $\{\alpha_1, \alpha_2, \ldots, \alpha_P\}$. Thus, the number of edges in the path is equal to $P-1$. Each pair of consecutive nodes (α_i, α_{i+1}) will be evaluated if the turning is compatible with the real movement with an error lower that ± 10 degrees. The estimation of weight is given by:

$$W_{\alpha_1,\alpha_P} = \frac{1}{2^P}(f_0 + \prod_{i=N_1}^{N_2} f_i e^{-\gamma_1 d_{o_i}}) \cdot \prod_{j=1}^{P-1}(f_j e^{-\gamma_2 d_{\alpha_j}} + \prod_{i=1}^{N_j} f_i e^{-\gamma_1 d_{o_i}}), \quad (1)$$

where N_j is the number of objects in the edge that connects the nodes α_j and α_{j+1}, the terms d_{α_j} represent the absolute distance differences between the annotated nodes and detected nodes distances whereas the terms d_{o_i} are the absolute distance differences between the annotated objects and the detected objects. The distance in the computation of d_{o_i} is only considered if there is correspondence between an annotated object and a detected object and both are of the same category. For objects without correspondence the distance d_{o_i} considered is zero and in this case only the parameter f_i comes into play. The parameters γ_1 and γ_2 are constants to weight the distance differences between objects and nodes, respectively. Finally, f_j and f_i represent penalty factors, which will be described below.

The first part of the Equation (1) considers that the first node in a movement may be not detected because the movement has begun in the middle of the edge that connect the initial node α_0 with the next α_1. In this case, the sequence N_1 to N_2 represents the list of objects considered in the first not complete edge travelled (both the detected and the annotated). The constant 2^P ensures that the weight W_{α_1,α_P} is within the interval [0,1]. The penalization factor f_0 quantifies the discrepancy between the detected (Td_i) and annotated node types (Ta_i) as:

$$f_0 = \begin{cases} 1 & Td_i == Ta_i \\ 0.5 & Td_i! = Ta_i \quad and \quad Td_i! = EN \quad and \quad Ta_i! = EN \\ 0 & Td_i! = Ta_i \quad and \quad (Td_i == EN \quad or \quad Ta_i == EN) \end{cases} \quad (2)$$

Here, EN denotes an End-node and this type of node suffers a higher penalization because is more difficult to have an error in such node type.

As a particular case, if the first node of the first edge has been detected $(\alpha_0 = \alpha_1)$, we have $(f_0 + \prod_{i=N_1}^{N_2} f_i e^{-\gamma_1 d_{o_i}}) = 2$ and complete edges are considered in the second part of equation (1). So, equation (1) is reduced to:

$$W_{\alpha_1,\alpha_P} = \frac{1}{2^{P-1}} \prod_{j=1}^{P-1}(f_j e^{-\gamma_2 d_{\alpha_j}} + \prod_{i=1}^{N_j} f_i e^{-\gamma_1 d_{o_i}}) \quad (3)$$

The parameter f_j in the second part of Eq.1 and in Eq.3 has a similar purpose to f_0 and its evaluation is given by:

$$f_j = \begin{cases} 1 \quad Td_j == Ta_j \quad and \quad Td_{j+1} == Ta_{j+1} \\ 0.125 \quad Td_j! = Ta_j \quad and \quad Td_{j+1}! = Ta_{j+1} \\ 0.35 \qquad\qquad others \end{cases}, \tag{4}$$

where T_{d_j} is the detected node type for $j-th$ node and T_{a_j} the annotated one. When both ends have the same type $f_j = 1$ and if both are different $f_j = 0.125$.

Finally, the parameter f_i quantifies the discrepancy between the detected (Cd_i) and annotated (Ca_i) object classes. For each non correspondence of class we set a penalty equal to 0.8, so 0.8^{N_C} where N_C is the number of objects with no correspondence.

$$f_i = \begin{cases} 1 \quad Cd_i == Ca_i \\ 0.8 \qquad other \end{cases} \tag{5}$$

In the implementation, the reference node must be taken into account for the annotated object because in the detected path the reference node is always the node where the robot came.

4 Results

In this section, the evaluation of the entire framework is presented. We have worked in a structured building distributed over four floors, which in turn are subdivided into four departments or areas. Figure 5(a) represents the occupancy map of the considered scenario, which corresponds to one of the four departments and covers an area of approximately 25×25 meters. The information included in the occupancy map of Fig.5(a) is only presented for clarification. Objects here are depicted with a certain colour according to their category. In addition, nodes of the topological map are referenced by an index and represented by red square points. The corresponding topological map is shown in Fig.5(b), where sub-nodes corresponding to objects are omitted. As zoom of the edge between nodes 2 and 3, Fig.5(c) represents the topological map with the annotated positions of the objects as sub-nodes of the edge.

In order to construct the reference topological map as input to the localization algorithm, only the distance from objects to reference nodes and distance between nodes are annotated. As as example of annotation of our test scenario, Table 1 includes the corresponding information for the edge connecting nodes 4→5 according to the nodes defined in 5(a). Each edge needs store its length, the node types that it connects and the objects that can be seen while travelling it. This information is encompassed in the following fields:

- Edge index: identifies the edge.
- Visualization flag: indicates if the object is visible when travelling from reference node to second one ('1') or viceversa ('-1'), and '0' indicates that the object is visible in both ways.
- Sub-node type: references the category of the object ('door', 'window', ...)

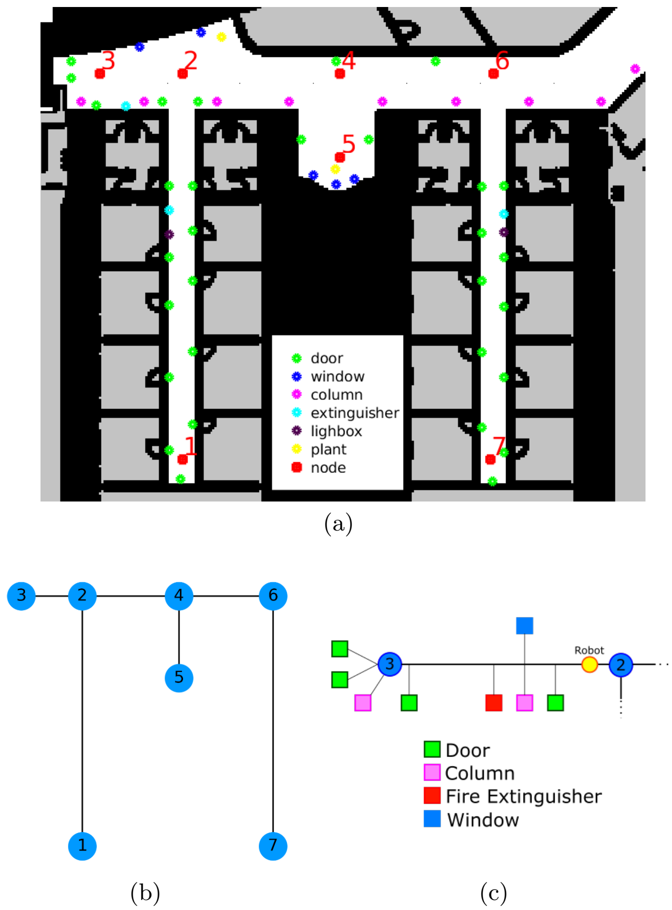


Fig. 5. Maps of the test scenario. (a) Occupancy map. (b) Topological representation. (c) Objects associated on the Edge between Nodes 2-3.

- Reference node: this information is included in order to get the distances to the reference node.
- Distance d_n: distance from the reference node to the normal projection of the object onto the edge line (see Fig.4). If the object is before (in the way back after the reference node) the distance is negative.
- Distance d_e: perpendicular distance from the object to the edge line (see Fig.4). This value is positive when the object is on the left with respect to the robot movement from the reference node and negative in the other case. This parameter has not been used so far in this work.

Edge index	Visual. flag	Sub-node type	Reference node	Distance d_n (m)	Distance d_e (m)	Detected class	Detected d_n (m)
2	0	'door'	4	3	1.87	'door'	15.70
2	0	'door'	4	3	-1.87	–	–
2	1	'window'	4	4.77	1.22	'window'	2.26
2	1	'window'	4	5.22	-0.19	'window'	4.66
2	1	'window'	4	4.77	-0.67	'window'	4.51
2	1	'plant'	4	4.4	-0.19	'plant'	2.71
2	1	–	4	–	–	'plant'	2.40
2	1	–	4	–	–	'plant'	15.70
2	-1	'door'	4	-0.55	-0.22	–	–

Table 1. Example of annotation/detection of edge objects to build the topological map regarding to the Fig.5(a).

By inspection Table 1, we can observe that according to the annotation there are two doors visible in both directions, three windows and one plant that are only visible from reference node to second node and a door that is only visible in the direction from second node to reference node. In the last two columns we report the objects detected by the system with respect to the annotation and their estimated distance d_n when the robot has followed the path 4→5. Note that perpendicular distance from objects to edges has not been considered in this work and will be estimated in future works. As we can observe, the second door is not detected by YOLO and the plant is detected as three different objects due to certain disengagements of the tracking algorithm. Despite some errors in the detection and tracking algorithms, as we will see later, the algorithm is still able to locate the robot.

In order to validate the process of estimating the pose of objects, we consider the triangulation of each pair of consecutive detected objects projected onto the ground plane. The x-coordinates of both image points of the centroids of each object detected by YOLO ($\boldsymbol{x}$ and $\boldsymbol{x}'$) allows to determine easily the angles β_1 and β_2 (see Fig.4) from the camera's angle of aperture α. The intersection of coplanar rays back-projected from ($\boldsymbol{x}$ and $\boldsymbol{x}'$) intersect at the object position O. In this model, we assume the validity of the pin-hole model. As an example, Fig.6

shows two consecutive captured images, where the x-coordinates of the detected fire extinguisher have been translated to their corresponding angles. According to the encoder readings, the pose estimation for the extinguisher relative to the first robot pose is: $x = 3.95$ m and $y = 0.62$ m, whereas reference estimation by manual measurements is: 3.98 m and 0.63 m. Unfortunately, pose estimation is not always so accurate due to several factors such as the detection provided by YOLO does not always match the bounding box of the object and noise in the measurement of encoders.

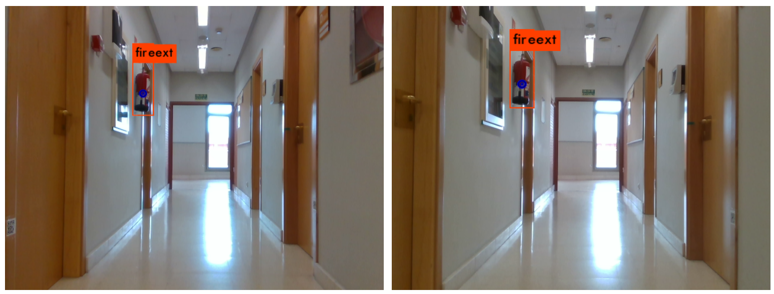


Fig. 6. Example images for fire extinguisher pose estimation (0.5 m ahead).

Table 2 describes the process of evaluation of the location function, which is defined by Eq.3 for the case in which the first node of the edge has been detected. Specifically, the reported example corresponds to the path 4→5, whose annotation and detection data are summarized in Table 1 considering the value of visualization flag equal to 1 according to the edge direction. For this example, we have $P = 2$, since there is only one edge with two terminal nodes, and the number of objects (annoted and/or detected) is $N = 8$, where 5 of them showed a correspondence between annotated and detected objects. Here, the classification of both terminal nodes (node 4 as Node-T and node 5 as an End-node) is correctly performed and, in consequence, $f_{j=1} = 1$. Contributions of each object with correspondence are summarized in columns 2 to 6 and the last column includes the contribution of unmatched objects. Based on results of the different sequences, we have adjusted the values of the parameters γ_1 and γ_2 in Eq.3, respectively, to 0.005 and 0.1. On the basis of these contributions, the final weight obtained for the analyzed edge is $W_{4\rightarrow 5} = 0.72$.

Nodes	Object 1	Object 2	Object 3	Object 4	Object 5	$N_c = 3$
$f_{j=1} = 1$	$f_{i=1} = 1$	$f_{i=2} = 1$	$f_{i=3} = 1$	$f_{i=4} = 1$	$f_{i=5}=1$	–
$d_{n_{j=1}} = 0.21$	$d_{o_{j=1}} = 12.70$	$d_{o_{j=2}} = 2.51$	$d_{o_{j=3}} = 0.56$	$d_{o_{j=4}} = 0.26$	$d_{o_{j=4}} = 1.69$	0.8^3

Table 2. Example of evaluation of the location function for the edge 4→ 5.

To test the behaviour of the proposed model, several sequences in our scenario have been analysed, where each one of them correspond to a single edge in the reference topological map. Each edge is defined from the path that goes from the source node to the destination node. Thus, we have the following sequences: sequences 2 $\rightarrow$ 3, 3 $\rightarrow$ 2 and 4 $\rightarrow$ 5. Each capture encompasses the frame captured by the camera and encoder information considering that the increase in distance between two consecutive catches is approximately 0.5 m. Images are processed by YOLO detector and as examples we can see in Figs.7(a), 7(b) and 7(c) the resulting detections for the first images of the three mentioned sequences. The tracking algorithm is run to associate multiple detections of the same object throughout the sequence. In Figs.7(d), 7(e) and 7(f) we can observe the object tracking flow in blue for the last image of each sequence. Here, each point represents the coordinates of the object in the sequential images during the sequence.

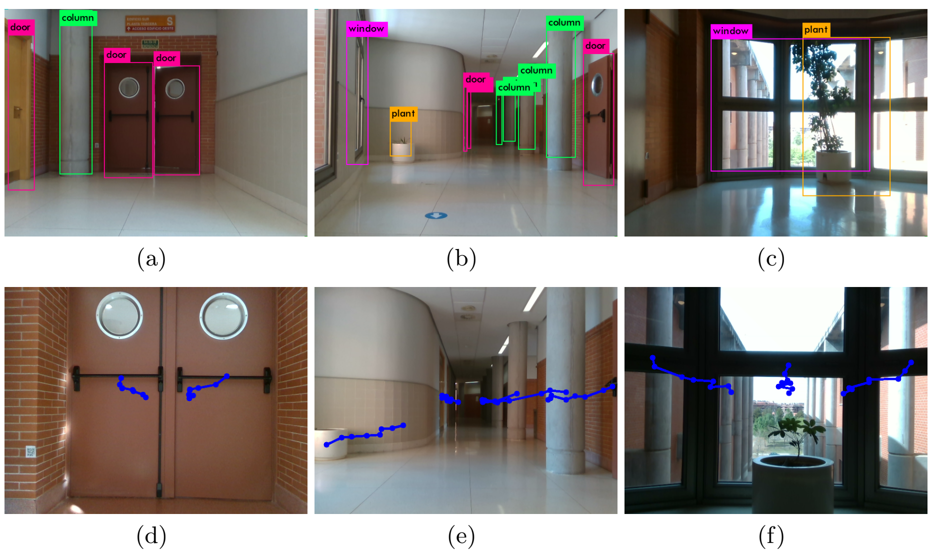


Fig. 7. Examples of results in the test scenario for the following sequences: 2 $\rightarrow$ 3, 3 $\rightarrow$ 2 and 4 $\rightarrow$ 5. (a,b,c) Visual object detection at the output of YOLO network for the first images of the sequences. (d,e,f) Tracking flow of detected objects in the last images of the sequences.

Table 3 summarizes the results for the three captures above mentioned. Each one of them is defined by the edge from source node to destination node. In addition, we report the edge corresponding to the highest weight that the system provides in the evaluation of location function and the edge with the second highest score. Despite some losses in object tracking, we can conclude that the algorithm is able to correctly estimate the robot location in all cases.

Edge	Detected Edge	Highest Weight	Closest to the Detected Edge	Second Highest Weight
$2 \rightarrow 3$	$2 \rightarrow 3$	0.69	$4 \rightarrow 5$	0.61
$3 \rightarrow 2$	$3 \rightarrow 2$	0.40	$4 \rightarrow 6$	0.38
$4 \rightarrow 5$	$4 \rightarrow 5$	0.72	$2 \rightarrow 3$	0.53

Table 3. Results of evaluation with the location function.

5 Conclusions

A model has been proposed in this work for automatic robot localisation based on the structure of the topological map of indoor environments. The evaluation of the environment is based on a localisation function that compares the relative position of objects and terminal node types detected by the system with respect to the reference map annotation. The results of the experiments show that the present method constitutes an efficient and accuracy system as a first approach despite certain limitations of the object detection and tracking algorithms.

For future works, the system should integrate depth information from RGB-D images to place objects in the topological map. Thus, the perpendicular distance from objects to edges will considered in the location function in order to give more accuracy results. In addition, other tracking algorithms will be tested in order to improve the robustness of the system. Finally, the complete system will be tested over longer routes in larger spaces.

Acknowledgements

This work was supported by the projects: (a) AIRPLANE (PID2019-104323RB-C31) and b) AAI (PID2019-104323RB-C32) funded by MCIN/AEI/ 10.13039/ 501100011033; and c) project CM/JIN/2019-022 of the University of Alcalá.

References

1. Simhon, S., Dudek, G. A global topological map formed by local metric maps. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications, pp. 1708–1714. IEEE (1998).
2. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788 (2016).

3. Pereira, R., Carvalho, G., Garrote, L., Urbano J. Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics. Applied Sciences, **12**(1319), 1–18 (2022).
4. Li, R., Jia, Z., Yang, Y., Gan, S. SORT with Depth Image Based Pedestrian Tracking Robots. In: Proceedings of the 33rd Chinese Control and Decision Conference (CCDC), 2021, pp. 1852–1857, (2021).
5. Cortes, C., Vapnik, V.. Support vector machine. Machine learning **20**(3), 273–297 (1995).
6. Levitt, T.S., Lawton, D.T. Qualitative navigation for mobile robots. Artificial Intelligence **44**, 305–360 (1990).
7. Barber, R., Crespo, J., Gómez, C., Hernández A., Galli, M. Mobile robot navigation in indoor environments: Geometric, topological, and semantic navigation. In Applications of Mobile Robots; IntechOpen: London, UK (2018).
8. Fuqiang, G., Xuke, H., Milad, R., Debaditya,A., Kourosh, K., Shahrokh, V., Jianga, S. Indoor localization improved by spatial context—A survey. ACM Computing Surveys (CSUR), **52**(3), 1–35 (2019).
9. Morar, A.,Moldoveanu, A., Mocanu, I., Moldoveanu, F., Radoi, I.E., Asavei, V., Gradinaru, A., Butean, A. A comprehensive survey of indoor localization methods based on computer vision. Sensors, **20**(9), 2641 (2020).
10. Kielar, P.M., Biedermann, D.H., Kneidl, A., Borrmann, A. A unified pedestrian routing model combining multiple graph-based navigation methods. In Traffic and Granular Flow'15, pp. 241–248. Springer, Cham, (2016).
11. Cheng, H., Chen, H., Liu, Y. Topological indoor localization and navigation for autonomous mobile robot. IEEE Transactions on Automation Science and Engineering, **12**(2), 729–738 (2015).
12. Chen, Kevin and Chen, Junshen K and Chuang, Jo and Vázquez, Marynel and Savarese, Silvio. Topological planning with transformers for vision-and-language navigation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.
13. Devendra, S.C., Ruslan, S., Abhinav, G, Saurabh, G. Neural Topological SLAM for Visual Navigation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
14. Blochliger, F., Fehr, M., Dymczyk, M., Schneider, T., Siegwart, R. Topomap: Topological mapping and navigation based on visual slam maps. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3818–3825, (2018).
15. Gunathillake, A., Huang, H., Savkin, A. V. Sensor-network-based navigation of a mobile robot for extremum seeking using a topology map. IEEE Transactions on industrial informatics, **15**(7), 3962–3972 (2019).
16. Lafuente-Arroyo, S.; Maldonado-Bascón, S.; López-Sastre, R.J.; Molina-Cantero, A.J.; Martín-Martín, P. LIDAR Signature Base Node Detection and classification in graph topological maps for indoor navigation. Proceedings of the Iberian Conference on Pattern Recognition (IbPRIA), 2022.