Journal of Intelligent & Robotic Systems (2021) 101: 32
https://doi.org/10.1007/s10846-020-01298-7

Semantic Active Visual Search System Based on Text Information for Large and Unknown Environments

Mathias Mantelli[1] · Diego Pittol[1] · Renan Maffei[1] · Jim Torresen[2] · Edson Prestes[1] · Mariana Kolberg[1]

Received: 2 September 2020 / Accepted: 10 December 2020 / Published online: 23 January 2021

Abstract

Different high-level robotics tasks require the robot to manipulate or interact with objects that are in an unexplored part of the environment or not already in its field of view. Although much works rely on searching for objects based on their colour or 3D context, we argue that text information is a useful and functional visual cue to guide the search. In this paper, we study the problem of active visual search (AVS) in large unknown environments. In this paper, we present an AVS system that relies on semantic information inferred from texts found in the environment, which allows the robot to reduce the search costs by avoiding not promising regions for the target object. Our semantic planner reasons over the numbers detected from door signs to decide either perform a goal-directed exploration towards unknown parts of the environment or carefully search in the already known parts. We compared the performance of our semantic AVS system with two other search systems in four simulated environments. First, we developed a greedy search system that does not consider any semantic information, and second, we invited human participants to teleoperate the robot while performing the search. Our results from simulation and real-world experiments show that text is a promising source of information that provides different semantic cues for AVS systems.

Keywords Semantic information · Active search · Visual search problem

1 Introduction

Searching for objects or regions of interest (ROI) in large-scale environments is a daily activity for human beings, in which visual recognition is a necessary and essential skill [14]. In this type of real situation, it is not prudent to assume that the desired object or ROI, is always present in the human's field of view since the beginning. The environment navigation is also a constant requirement of life for humans, and it must be as intelligent as possible, even in unknown places [32]. In such cases, humans must actively navigate and search for objects in the environment, relying on their visual recognition abilities [14, 30].

Similar to humans, autonomous robots are also dealing with tasks related to object searching, such as home assistance, delivery of packages, manipulation in factories, and fetch and carry. This progress is possible thanks to the late advances in the field of mobile robotics, more specifically in the localisation, mapping, navigation, and exploration ones [5]. Similar to humans in the context of object searching tasks, robots can also not rely on the assumption that objects are already within their field of view. Hence, they have to find objects in large-scale environments based on primarily visual sensors, which is known as an active visual search (AVS) problem [5, 30], and the main problem addressed in this paper. In more details, an AVS aims to calculate a set of sensing actions to

✉ Mathias Mantelli
mfmantelli@inf.ufrgs.br

Diego Pittol
dpittol@inf.ufrgs.br

Renan Maffei
rqmaffei@inf.ufrgs.br

Jim Torresen
jimtoer@ifi.uio.no

Edson Prestes
prestes@inf.ufrgs.br

Mariana Kolberg
mariana.kolberg@inf.ufrgs.br

[1] Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil

[2] Institute of Informatics, University of Oslo, Oslo, Norway

bring the target object into the mobile robot's sensor field of view.

The search strategy is one of the critical factors of an AVS system since it directly impacts the efficiency of the system [5]. It is responsible for maximising the probability of detecting the target object in the environment and minimising the total cost of the task [30]. There are different approaches to measure this cost, and the most common options are the time or distance travelled, and the movements of the robot (each type of movement has a different value). For example, imagine that a courier robot is responsible for delivering a package to a specific room within a large-scale and unknown building. The most straightforward search strategy would be the robot visiting all places in the environment, and visually checking whether every new room is the target one. Even though many robot sensors have a limited field of view, it is very inefficient and time consuming to make the robot visit all the existing rooms of a building to accomplish its task. In contrast to this simple strategy, semantic information about the environment could be collected and used by a more efficient planner. Therefore, instead of making the robot to exhaustively visit all the existing rooms in a building to deliver the package at the target room, the search strategy would be able to reason over the semantic information and extract important cues to improve the searching.

The research community has proposed essential and useful works related to AVS problem [5, 15, 25, 30, 31]. However, despite these contributions, the problem is proven to be NP-Complete [33, 37]. Then, the optimal search solution can be computed by approximation [30], minimising the search cost as much as possible. In the example of the courier robot previously introduced, taking advantage of semantic information of the environment provides search cues for this approximation. Then, the robot should be able to reason over their sensor readings, infer new knowledge, and increase their level of abstraction of the environment over time [8].

However, making the robot able to acquire such information from real scenarios, i.e. unexplored environments, includes additional cost and increases the difficulty level of AVS systems [5]. The challenge of such systems relies on balancing exploration and knowledge exploitation, i.e. should the robot explore further or search for the target in the already known regions. In our work, a courier robot is responsible for delivering a package to a specific room in a large-scale and unknown environment, with the shortest possible distance travelled. Our searching strategy relies on numbers visually extracted from door signs in the environment, as any information is provided to the robot beforehand.

The contributions of our paper are summarised in two parts. First, it presents a semantic AVS system that efficiently searches for objects, which in our case, the target is represented by the number of door signs, in large-scale and unknown buildings. Second, it also presents a quantitative analysis of our semantic AVS system in four different simulated indoor environments and one physical test as proof-of-concept implementation in which a robot is tasked with finding a target door sign. Besides, it compares the performance of our system with a purely geometric AVS system called Greedy, as well as with humans teleoperating the robot to perform the same search task.

Different object searching approaches have been proposed by the research community as earlier presented. In many of them, the robot is tasked with finding objects based on their visual appearance or position in the environment (e.g. a plate on the kitchen table). However, to the best of our knowledge, AVS systems that search for door signs (not areas in general) and take numbers as input to their strategy are not well explored yet. Hence, the novelties of our paper in comparison to the already published systems are threefold.

First, the target of our proposed semantic-based AVS system is a specific door sign (called *goal-door* in this paper) within an unknown indoor environment, instead of an ROI, chair, table, or kitchen utensil like others works. This novelty enables, for example, autonomous courier robots to perform the final part of the delivery task, which happens after the robot arriving at the buildings, where there is no map available. Our system is relevant in moments such as COVID-19 pandemic, in which people are recommended to stay at home and avoid social contact. Hence, it stimulates the use of fully autonomous robots for performing the entire delivery task.

Second, our semantic AVS system relies on textual information as a visual cue, and more specific numbers, extracted from the door signs. Large buildings, for instance, are divided into many small rooms, and usually comply with a pattern of signing each room [1, 4, 18, 34]. Using numbers is different from considering the size of an ROI, or the features and colours of an object, for instance. If we could analyse humans while looking for a door sign in an unknown environment, most of them would try to figure out the door signing pattern. They would avoid exploring the entire building by analysing how the door signs are related to each other to infer whether the current corridor is promising in terms of containing the goal-door. A courier robot could behave in a similar way to efficiently perform the searching task. Hence, our AVS system processes the numbers to infer semantic information from them, such as odd and even characteristics, and whether the sequence of numbers is increasing or decreasing. In addition to the semantic information, our system also takes in consideration geometric information, which is the distance between the robot's pose and the unknown regions, and the history of the robot's orientation.

Third, our paper also presents an analysis of the advantage of using text from the door signs and its inferred semantic information as input to our semantic AVS system, instead of limiting it only to geometric information. The combination of this semantic information, inferred from the numbers, and the geometric information, extracted from the environment, are useful inputs for the reasoning of our system. It permits the efficient computation of search and exploration steps, guiding the robot towards areas more likely to lead to the target room. Lastly, in addition to this computation, which is fully probabilistic, our system also builds a 2D map of the environment. It is segmented based on spatial density information, i.e. according to different sizes of free space [20]. Our system takes into consideration the laser readings to build the 2D map, and the images of an RGB camera to extract the numbers from the door signs. The numbers are used to infer semantic information and search cues, whereas the map indicates new regions to explore, and when combined, they indicate which direction is more likely to contain the goal-door to guide the robot.

The remainder of our paper is organised as follows. After reviewing the literature in Section 2, Section 3 describes our semantic AVS system and its basic components. Section 4 explains our semantic planner and how it considers the door signs as exploration cues to reach the goal-door. Next, Section 5, introduces the experimental setup, and then it compares the results of our semantic AVS system to, first, a purely geometric and coverage-based AVS system called Greedy, and second to human participants performing the same task using the robot embodiment (teleoperating the robot and observing its sensor readings in the simulation setup). Lastly, it presents the results of our semantic AVS system in a proof-of-concept that uses a physical robot performing the search in an unknown environment. The paper conclusion is presented in Section 6, discussing the demonstrated outcomes.

2 Related Work

Visual object search is a problem that has been studied for many years in the robotics field. The proposed approaches range from multi-agent collaborating to search for an object [36], to a single robot actively performing a semantic-based search [38]. After many years subtopics of research arose within the object search, such as Indirect and Active Visual searches. Despite this long period in which new approaches have been proposed, there are no detailed surveys in the literature shedding light on this latter subtopic. However, it is possible to find comprehensive surveys on wider topics such as salient objects detection [11], visual attention [9], and as pointed out in [5], active vision [12, 13]. It is important to mention that even though it has not been proposed as a survey, in [5] Aydemir et al. presented a comprehensive review of some of the most important works related to AVS. Hence, we review other works not presented in [5], that are as important to the development of our paper as the presented ones. This review allows us to show how our work compares to the visual object search body of research. It also shows how our system contributes to the advancement of state of the art.

In a series of papers, Aydemir and colleagues presented spatial representations and a different planner to the AVS problem. In [7], the authors proposed a spatial relation that describes topological relationships between objects. They used that description to create potential search actions for the AVS problem since they aimed to relax the assumption that objects start within the robot's sensory reach. Their spatial representation was improved in [6], in which they proposed a combination of a 3D metric map, which supports obstacle avoidance and path planning, and a topological map called place map, which maintains the topology of the environment. The outcome of such combination was a conceptual map, which connects symbols representing instance knowledge about the environment with spatial concepts such as objects, room categories or appearances. The spatial relation introduced in [7] was used later in [2] when they combined semantic cues to guide the object search process in a larger environment. A switching planner combines a continual classical planner, which decides the overall strategy of the search, and a decision-theoretic planner, which uses a probabilistic sensing model to set the low-level observation actions. In this same paper, they also proposed an exploration strategy which considers the object search task, since their start without an initial map of the environment. The next proposal has argued that there is a strong correlation between local 3D structure and object placement, which is called the 3D context. The authors argued in [3] that local 3D shape around objects is a strong indicator of the placement of these objects. Hence, they used a more general model to learn the relationship between 3D context and objects, in contrast to the correlation between objects and the appearance of the environment. The evaluation of their approach was performed considering a large RGB-D dataset, showing the effect of using 3D context in an object detection task. Besides making an RGB-D dataset publicly available in [3], in [4] Aydemir et al. also published a dataset called KTH. In this case, the dataset is composed of a set of floor plans that encompasses, in total, 37 buildings, 165 floors and 6248 rooms. In addition to KTH, another contribution of their work was two methods for predicting indoor topologies and room categories given a partial map of the environment. The goal was to predict what lies ahead in the topology of the environment through its topology. Finally, in [5] the AVS is performed without any initial map, and hence, besides performing the search, their approach explored the environment as well. This was one of their main contributions, that is the balance between exploration and exploitation, which makes the robot explore more regions of the environment only after carefully searching for the object in the known regions. Their proposed AVS system reasons about whether exploit the known part or explore the unknown part, based on a model that describes the distribution over possible extensions to the current world.

The idea of relying on significant and visible landmarks to narrow down the search was not found only in [3]. Zeng et al. exploited background knowledge about common spatial relationships between landmarks and target objects [38]. Their proposal, called Semantic Linking Maps (SLiM), maintained the belief over the locations of the target object and the landmark. Simultaneously, it accounted for probabilistic inter-object spatial relations. In contrast to the 3D context-based AVS systems, Rasouli et al. proposed an attention-based AVS system [25]. They argued that an AVS system must be responsive, directive, spatiotemporal, and efficient, which are the characteristics addressed by their model. It embedded visual attention in an n-step decision-making algorithm formalised as a 1st-order Markov process. The use of visual attention increased the robot's awareness of the environment. Hence, they used all relevant visual information that was available, leveraging the spatial and appearance information about the object. Rasouli and Tsotsos also relied on visual attention methods to reduce computational costs on their robotic visual search [26]. They proposed a three-pronged probabilistic search algorithm that incorporated three forms of visual attention, that are viewpoint selection, saliency, and object-based models. On their model, the attention is used to generate maps with highlighted areas in the image which are more likely to contain an object of interest. The experiments showed that the proposed three-tier attention framework decreased the search cost in terms of distance travelled, search time, and the number of actions taken. Saidi et al. explored a different robot than the other works that opted for wheeled robots since their AVS system was proposed based on the specificities of a humanoid robot [28]. A visibility map, which constrains the sensor parameter space, was used to avoid unnecessary calls to the rating function, that evaluates the interest of a potential next view through the analysis of the theoretical field of view.

It is worth to mention that our semantic AVS system also considers the exploration of unknown environments as part of the problem. We aim to perform AVS in an entire unknown search space, which requires a way of switching between the exploration of unknown regions and the exploitation in already known regions. Hence, it is important to present some works related to exploration. Here, they are divided into two significant groups regarding their goals. First, strategies that aim to explore the whole environment, usually finishing when the robot has visited the entire free area [16, 24]. Second, strategies called goal-directed that aim to reach a goal, such as searching for an object, a room, or a person.

The papers reviewed in this section use semantic information to improve their findings. Some of them use a semantic map, whereas others use semantic properties from objects. The system proposed by Aydemir et al. [5] focuses on a large-scale environment, where the robot should find objects using mainly visual sensing. They affirm that rather than performing an exhaustive search in the area, their system could find the object guiding the robot towards to areas more likely to contain it. The probability is calculated considering extracted semantic cues from appearance, geometry, the topology of the environment, and general semantic knowledge of the indoor space. They showed that the results improved drastically after including a semantic description in their search system.

Differently, the framework proposed by Veiga et al. [35] searches for objects in domestic environments. It is composed of a system that perceives the query object in RGB-D images through an inference process and sensor information. The outcome of this process, called knowledge, is saved and updated in a semantic map. Experiments in a realistic apartment have shown that their framework worked well in practice, presenting a reliable and efficient search approach.

Another significant work that searches objects in domestic scenarios is Rogers' et al. [27]. In contrast to [35] that proposed a modular system, their approach considered the context of the environment. A graph, connecting places and objects within these places, is used to predict the presence (or absence) of objects based on the room categories. The reasoning made over the graph, combined with a planner, is used to perform an object search task. Experiments showed that the robot was able to find objects in the environment.

Talbot et al. [32] and Schulz et al. [29] proposed navigation approaches that are also Goal-Directed, despite not being exploration ones. The idea of an original and abstract map that links symbolic spatial information with observed symbolic information and actual places in the real world was firstly introduced by [29]. This map is used to make inferences about the location of places. Later, Talbot et al. [32] extended the idea of the abstract maps, proposing a novel method that defines the topological structure and spatial layout information encoded in spatial language phrases. The system has shown to complete the task by travelling slightly further than the optimal path.

Despite the good outcomes from the solutions presented by the papers mentioned above, there is still room for improvements. In [5] Aydemir et al. depended on prior semantic knowledge about indoor spaces obtained from databases. Talbot et al. [32] and Schulz et al. [29] depended on a priori abstract maps. Veiga et al. [35] required beforehand information to learn about objects and the environment. Additionally, it used a 3D recognition based framework from the Point Cloud Library (PCL) for object recognition, which is computationally expensive. Rogers et al. [27] also implemented PCL to segment data from RGB-D sensor, continuing to cluster the points, what is a heavy workload for computers. It is also important to highlight that none of them has explored the benefits of textual information available in the environment.

Our proposed AVS system reads the door sign numbers through an efficient computer vision algorithm and analyses them to decide whether the current path is promising for the robot to find the goal-door. It does not require an environment description nor other instruction in advance, which is suitable for tasks in unknown environments. Additionally, it is not computationally expensive, and a simple computer and two cameras embedded in a robot can execute it. Relying on textual information from the environment, and from that infer semantic information, is another contribution of our work in comparison with the other papers reviewed here. Given that no information or map is necessary as a requirement for our system, it is a good solution for entirely unknown environments.

3 Semantic-Based AVS System

This Section details the basic modules that compose our semantic AVS system. It starts with an overview of the system in Section 3.1. Then it goes to the Mapping module in Section 3.2, to explain how the 2D grid map is built. The Map Segmentation module is introduced in Section 3.3, which presents how the map is split into different segments. Finally, Section 3.4 describes how the numbers are extracted from the door signs through computer vision algorithms.

3.1 Overview

Our proposed system performs an AVS, i.e. it guides the robot through an unknown environment until the robot reaches a specific location or finds an object. Our system focuses on indoor environments, such as buildings with many rooms identified by door signs. One of its advantages is that it does not require any a-priory knowledge about the environment, such as the door signs arrangement or where the robot should be heading. In our paper, the target to be found is a door sign (called here as goal-door), which is identified by a number. An example to illustrate the usage of our system is a courier robot that delivers a package. From the restaurant until the destination building, it uses Google Maps and its embedded GPS to navigate through the city. However, once it is inside the building, it does not have a map to plan its path through the corridors. Therefore, the robot has to search the target goal-door to deliver the package to the customer.

Our system is composed of four modules: Mapping, Image Processing, Map Segmentation and Semantic Planner. The Semantic Planner module, presented in Section 4, is the main contribution of our paper. It requires a base system to work, composed by the first three other modules, that are discussed in Sections 3.2, 3.3, 3.4. The first of these three modules, Mapping, aims to build a 2D grid map of the environment using the Histogramic In-Motion Mapping (HIMM) technique [10], that takes as input the readings from a 180° laser sensor. The next module, Image Processing, processes the images taken by two RGB cameras, and it analyses them to recognise the number from door signs. Once identified, the module includes them into the 2D grid map at their respective side, i.e. left or right wall. The third module, Map Segmentation, is responsible for segmenting the free space of the 2D grid map according to its size using the Kernel Density Estimation (KDE) approach introduced by Maffei et al. [20]. This module also assigns to each segment of a corridor its respective list of door signs, that is the list of numbers recognised while the robot was within the corridor. The last module, Semantic Planner, calculates which path is more likely to contain the goal-door given its detected doors. The Boundary Value Problem (BVP) [23], calculated over the grid map and the Voronoi diagram [17], moves the robot towards the path that is most attractive, defined by the Semantic Planner.

3.2 Mapping Module

As the robot, equipped with a laser range-finder, moves through the environment, it reads a set of measurements that are used as input to the HIMM method in the Mapping module. It aims to build a 2D occupancy grid map $\mathbf{M}$ of the environment, Fig. 1a. Over $\mathbf{M}$, the Mapping module also computes the Voronoi diagram to have the centre cells of the free spaces, represented by the green lines in Fig. 1b. The yellow region represents the free space that was visited by the robot kernel, i.e. the circle centred at robot's pose. Based on that, the BVP smoothly moves the robot through the environment, avoiding obstacles and keeping it as close as possible to the Voronoi cells.

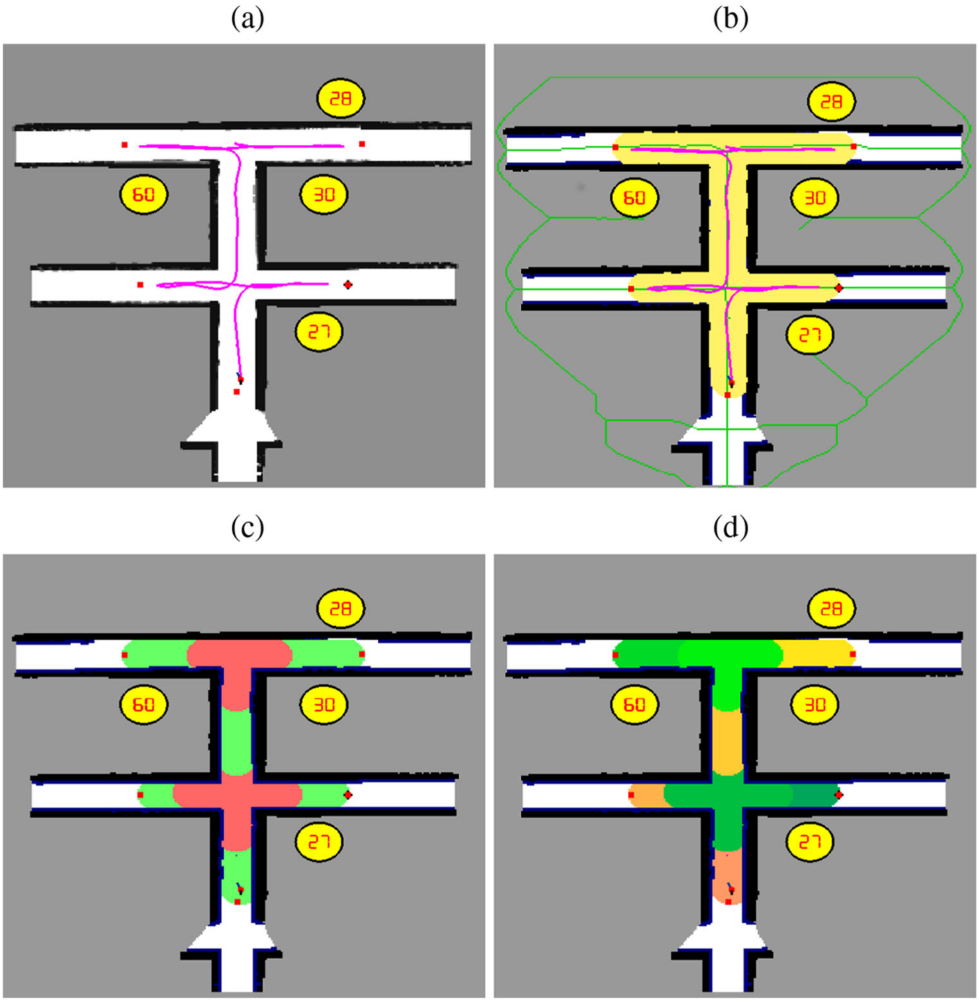


Fig. 1 Example of our mapping and segmentation modules. (**a**) shows the white area representing the free cells and the pink line the robot path, (**b**) the marked yellow areas representing the visited cells considered by the kernel of Eq. 3, and the Voronoi through the green lines, (**c**) our KDE-based module segmenting the visited cells of (**b**) as two types, and (**d**) the segment identification using different colours

3.3 Map Segmentation Module

The Segmentation and Mapping modules are executed simultaneously, aiming to split the free space of $\mathbf{M}$ into multiple regions according to the size of free areas. Every segmented region is called a segment, and it is used to store important information from this group of cells. In $\mathbf{M}$, there might be different types of segments. Figure 1c illustrates the case in which the Segmentation module considers only two types. In this case, it means that all green segments, or all the red ones, have a similar size of free space computed using the KDE. Besides their type, each segment is also singularly identified, such as $\mathbf{s}_i$ for the i-th segment within $\mathbf{M}$. Figure 1d shows the segmented 2D map as if each segment was identified as one colour.

For this purpose, the Segmentation module uses the KDE approach [20]. The $K(\cdot)$ is a uniform kernel that computes the size of the free area covered by it, defined as

$$K(d) = \begin{cases} a & \text{, if } d \leq r \\ 0 & \text{, otherwise,} \end{cases} \tag{1}$$

where r is the radius and a is the height of $K(\cdot)$, and d is the Manhattan distance from the current cell being measured, $\mathbf{c} \in \mathbf{T}$, to the centre of the kernel, cell $\mathbf{c}_k \in \mathbf{M}$. $\mathbf{T} \in \mathbf{M}$ is a subset of cells that have been within the area of the kernel in any moment. For a given cell $\mathbf{c}$, $Q(\cdot)$ tests whether it is free, and it is defined as

$$Q(\mathbf{c}) = \begin{cases} 1 & \text{, if } \mathbf{c} \text{ is a free cell} \\ 0 & \text{, otherwise.} \end{cases} \tag{2}$$

Combining the previous function into the KDE approach, it is possible to calculate the kernel density. For a cell $\mathbf{c}_k$, its free space $\Psi(\cdot)$ is computed by

$$\Psi(\mathbf{c}_k) = \sum_{\mathbf{c}}^{\mathbf{T}} Q(\mathbf{c}) K(\|\mathbf{c} - \mathbf{c}_k\|). \tag{3}$$

According to Eq. 2, when unknown cells are found within the kernel area, it is still possible to compute $\Psi(\cdot)$. Therefore, once unknown cells return zero from Eq. 2, $\Psi(\cdot)$ can differentiate density measures when obstacles are surrounding $\mathbf{c}$ and decrease the size of the area computed by the kernel.

Assuming that the Segmentation module considers different sizes of free areas, and given that Eq. 3 calculates the size of free area surrounding a cell $\mathbf{c}_k \in \mathbf{M}$, $\Psi(\cdot)$ can be used in the Segmentation module as

$$\Upsilon(\mathbf{c}_k) = \lfloor \Psi(\mathbf{c}_k)/\delta \rfloor \quad (4)$$

where δ is a threshold that defines how many different sizes of free areas are considered by the segmentation function, $\Upsilon(\cdot)$. Therefore, a high δ means Eq. 4 considers few different sizes, whereas a low δ is the opposite.

A segment $\mathbf{s}$ represents a group of free and adjacent cells from $\mathbf{M}$ that have the same $\Upsilon(\cdot)$. Figure 1d demonstrates different segments, in which each one has a different colour. For example, given $\mathbf{c}_0$ and $\mathbf{c}_1$ as two free and neighbouring cells in $\mathbf{M}$, and that $\Upsilon(\mathbf{c}_0) = \Upsilon(\mathbf{c}_1)$, then both belong to the same segment $\mathbf{s}_0$. Otherwise, a new segment $\mathbf{s}_1$ is created and $\mathbf{c}_1$ is associated to it. Thus, the segmentation of free adjacent cells from $\mathbf{M}$ is based on Eq. 4.

3.4 Image Processing Module

The last module that completes the basis of our semantic AVS system is the Image Processing one. It aims to recognise the number of a door sign that may be in an RGB image. The idea here is to use one well known existing text recognition algorithm [19, 22, 39], since this is not the focus of our paper, and any approach can be used. The chosen work is the one proposed by [22] due to its real-time recognition aspect, and its robustness against noise and low contrast of characters. Besides, it does not require any information or preparation beforehand.

For a given image $\mathbf{I}$ that was captured by the robot at cell $\mathbf{c}$, for example Fig. 2a and c, the image processing module returns a list $\mathbf{L}$ containing the recognised number from door signs. In the case of Fig. 2, $\mathbf{L}$ would contain only the number 228. Figure 2 also shows where the detected door signs are included into the 2D map $\mathbf{M}$. Figure 2c shows an image taken by the camera on the robot's right side, and hence, the number is included into the map at the same side, as shown in Fig. 2b. Given that the goal of signing rooms is to provide a unique door sign for each of them, it is assumed that there are not two door signs in a corridor identified by the same number. After receiving $\mathbf{L}$, it must be merged with the numbers of the door signs from the nearest segment of $\mathbf{c}$. For this process, it is important to define $S(\mathbf{c})$ as a function that returns the nearest segment of a cell $\mathbf{c}$, and $L(\cdot)$ as a function that returns the list of door signs from a segment. Thus, each door number $l \in \mathbf{L}$ is included in the list of door signs from the segment of $\mathbf{c}$, $l \cup L\big(S(\mathbf{c})\big)$. Besides, each l has an occurrence number, that increases by one every time that the image processing algorithm recognises it. If the robot revisits a place and recognises a l that already exists in $L\big(S(\mathbf{c})\big)$, then its occurrence number is summed to the one in $L\big(S(\mathbf{c})\big)$.

4 Semantic Planner

The previous Section 3 explained the necessary components that compose the basis of our semantic AVS system, i.e. the Mapping, Segmentation and Image Processing modules. The explanation continues with the Semantic Planner module, presenting how the planner decides whether the robot should continue its search to find the target, or change its path to a known region. To facilitate the explanation, imagine that the robot has partially visited the environment while running the necessary components of the AVS system. Then, the regions are mapped, segmented, and all the visited doors were recognised. Assuming the existence of such a map, aimed to help the Semantic Planner explanation, this section describes in details the planner.

Our semantic planner is composed of five different parts, in which two of them are semantic-based, Growing Direction and Parity, and the other three are geometric-based, Doors and Robot Orientations, and Distance. The combination of them leads to a planner that is neither exclusively semantic or geometric. This non-exclusivity characteristic is suitable for situations where the environment does not have semantic cues to be considered by our approach. All the five factors are presented individually in this section, introducing the semantic-based firstly, and then the geometric-based ones. However, as this section follows a top-down fashion to introduce the whole planner, the final formula that combines all the five factors is presented before them. Therefore, the reader can have a general idea of how the factors are used and later understand how they work.

4.1 Final Formula

During the searching process, our semantic AVS system analyses the environment while the robot has not found the goal-door. If it realises the current region of the environment is not promising, the system guides the robot to another direction. To decide the best frontier to go given the set of frontiers, for each candidate cell $\mathbf{c} \in \mathbf{C}$, the planner calculates its attractiveness factor $\varphi(\mathbf{c})$. This factor is the outcome of the combination of five factors briefly presented earlier. These candidate cells in $\mathbf{C}$ are the ones in the centre of the free space, i.e. in the Voronoi, and within a frontier, that is the boundary between visited and not visited cells. Graphically speaking, five candidate cells are shown in Fig. 1a, represented by the red dots near the pink line ends. In this case, $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_5\}$. The visited cells are the free cells that were within $\mathbf{T}$, represented by the yellow region in Fig. 1b, whereas the white region represents the

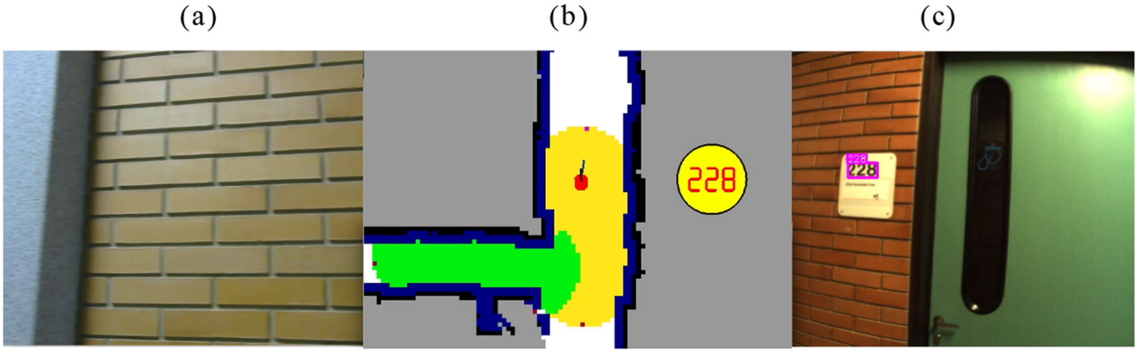


Fig. 2 Example of the Image Processing module processing two images, in which (**a**) is an image taken from the left camera, and (**c**) an image taken from the right camera. (**b**) shows the 2D map of the environment and the position of the door sign number 228

free space that is not close enough to the kernel centred at robot's position. The Final formula explanation is divided into two parts, in which the semantic factor is presented before the geometric factor. In the following subsections, the components of each factor, i.e. Growing Direction, Parity, Robot and Door Orientations, and Distance, are presented.

The semantic factor, $\mathtt{S}(\mathbf{c})$, combines the Growing Direction and the Parity factors, $\varphi_g(\cdot)$ and $\varphi_p(\cdot)$, respectively. The idea of the first factor is to return high values when the segment $S(\mathbf{c})$ is more likely to contain the goal-door given the door sign sequence. On the other hand, the second part of $\mathtt{S}(\mathbf{c})$ aims to analyse the parity of $S(\mathbf{c})$ and compare it to the goal-door parity. When the robot is in a $S(\mathbf{c})$ that is not likely to contain the goal-door, either due to $\varphi_g(\cdot)$ or $\varphi_p(\cdot)$, it should go to another path and continue the active search. Given that both Growing Direction and Parity factors are important, in $\mathtt{S}(\mathbf{c})$ they are multiplied by each other. If one is low, the result of $\mathtt{S}(\mathbf{c})$ will end up being low as well, even when the other is high. It is important to highlight that $\mathtt{S}(\mathbf{c})$ is completely probabilistic, and due to how both $\varphi_g(\cdot)$ and $\varphi_p(\cdot)$ are modelled, $\mathtt{S}(\mathbf{c})$ becomes robust to outliers. The Semantic factor is given by

$$\mathtt{S}(\mathbf{c}) = \varphi_g(\mathbf{c})\varphi_p(\mathbf{c}) \tag{5}$$

Differently, the geometric factor, $\mathtt{G}(\mathbf{c})$, multiplies the Robot and the Door Orientation factors, $\varphi_r(\cdot)$, $\varphi_o(\cdot)$ respectively, by the Distance one, $\varphi_d(\cdot)$, once the further they are, the less they matter. Then, the outcome of these multiplications is summed to the $\varphi_d(\cdot)$. The geometric factor is given by

$$\mathtt{G}(\mathbf{c}) = \frac{\big(\varphi_o(\mathbf{c}) + \varphi_r(\mathbf{c})\big)\varphi_d(\mathbf{c}) + \varphi_d(\mathbf{c})}{3.0} \tag{6}$$

Finally, in order to define the best $\mathbf{c} \in \mathbf{C}$, i.e. the $\mathbf{c}$ that is more like to contain the goal-door, each of them is submitted to the Eq. 7. Here, α is a threshold that controls the importance of the $\mathtt{S}(\mathbf{c})$ and $\mathtt{G}(\mathbf{c})$, and it ranges as $1 \leq \alpha \leq 0$. The outcome of Eq. 8 is the $\mathbf{c}_*$, that is the candidate cell in which its $S(\mathbf{c})$ is more likely to contain the goal-door,

$$\varphi(\mathbf{c}) = \mathtt{S}(\mathbf{c}) * \alpha + \mathtt{G}(\mathbf{c}) * (1.0 - \alpha) \tag{7}$$

$$\mathbf{c}_* = \underset{\mathbf{c}\in\mathbf{C}}{\arg\max}(\varphi(\mathbf{c})) \tag{8}$$

4.2 Growing Direction Factor

Usually, doors of buildings are signed in sequence and sorted (either increasing or decreasing order). For example, the first number of a corridor is smaller than the last one, or in the other way around. This characteristic can be inferred through the door sign sequence analysis. Imagine, for instance, that a robot is in a corridor where the number of the first door sign is larger than the goal-door one, and this corridor has an increasing door sign sequence. Hence, in terms of the Growing Direction factor, the robot should not consider this path as promising, once it is not very likely that its door sequence contains the goal-door. Therefore, the proposed Growing Direction factor, a type of semantic information inferred from the door sign sequence, is highly useful to our semantic AVS system, given that it indicates the door signs organisation in a segment.

For each $\mathbf{c} \in \mathbf{C}$, the Growing Direction factor first calculates the angle in which the door sign sequence is increasing, $\theta_i\big(S(\mathbf{c})\big)$. To determine it, all the detected door signs of the segment $S(\mathbf{c})$ are considered, $L\big(S(\mathbf{c})\big)$, as illustrated by Fig. 3a. Then, for all possible pairs of two different door signs, in which one is larger than other, the vector that connects them is computed. Figure 3b demonstrates an example for door sign number 1, and how the vectors are computed in pairs, such as (1,2), (1,3), (1,4), and (1,5). As it shows, the door sign number 1 has four vectors, while the door sign number 4 has only one. Just to illustrate, if we align all these detected door signs, as shown by Fig. 3c, it would be easier to understand that the sum of vectors from Fig. 3b and the final $\theta_i\big(S(\mathbf{c})\big)$, indicate that the sequence increases to the right. To make this process even clearer, Fig. 3d repeats the same procedure to the other door signs remaining, i.e. 2, 3 and 4.. Here, it is important to mention that the door signs, i.e. the yellow circles, were represented within the white area to help the explanation. In the simulator used in our paper, they appear within the grey area, as illustrated in Fig. 2b.

Figure 4 illustrates a partial map from the simulator used in our paper, and it helps to explain the importance of the Growing Direction. The robot has started at the intersection of three corridors, and it has chosen the number 3, i.e. the one on the right. According to the direction of the robot in this corridor 3, the door sign sequence is considered as increasing. Hence, in this current scenario, if the goal-door were 40, for instance, the Growing Direction factor would consider corridor 3 as promising. In contrast to this, the same corridor 3 would be not promising if the goal-door was 2, given that the sequence only increases meaning that the distance from door sign 2 also increases as the robot continues in that corridor. Besides these two examples, which help to understand how the Growing Direction factor behaves, there is a third case that is important to mention. Imagine that the goal-door is 21, this factor would be high until the door sign 20, but after that, its value would decrease as the robot continue in corridor 3 and the door sign sequence increases. Hence, just by the Growing Direction factor and regardless the parity of both the goal-door 21 and the door signs within the sequence, the robot should continue its search in either the corridor 1 or 2.

One possible solution to deal with the aforementioned third case is to measure whether all the door signs within the sequence are smaller or larger than the goal-door. Hence, the amount of door signs that are smaller or larger than the goal-door are counted by the functions $L^{<}\big(S(\mathbf{c})\big)$ and $L^{>}\big(S(\mathbf{c})\big)$, respectively. The factor $\zeta(\cdot)$ measures the possibility of a segment to have door signs smaller or larger than the goal-door, defined by

$$\zeta\big(S(\mathbf{c})\big) = \frac{\Big(L^{<}\big(S(\mathbf{c})\big) - L^{>}\big(S(\mathbf{c})\big)\Big)}{\max\Big(L^{<}\big(S(\mathbf{c})\big) + L^{>}\big(S(\mathbf{c})\big), w_g\Big)}, \tag{9}$$

where $-1 \le \zeta\big(S(\mathbf{c})\big) \le 1$, in which $\zeta\big(S(\mathbf{c})\big) = 1$ means that in $S(\mathbf{c})$ there are only larger door signs, $\zeta\big(S(\mathbf{c})\big) = -1$ only smaller door signs, and $\zeta\big(S(\mathbf{c})\big) = 0$ that both $L^{<}$ and $L^{>}$ are equal. w_g is a threshold used to control the minimum amount of detected door signs are necessary to this equation reaches 1 or -1.

In addition, Growing Direction factor also considers $\theta_f(\mathbf{c})$, that is the Voronoi angle at cell $\mathbf{c}$. The difference angle between $\theta_f(\mathbf{c})$ and $\theta_i\big(S(\mathbf{c})\big)$, measured by $\gamma(\theta_f(\mathbf{c}))$, indicates whether $\theta_f(\mathbf{c})$ is pointing to the same direction than $\theta_i\big(S(\mathbf{c})\big)$. Then,

$$\gamma(\theta_f(\mathbf{c})) = 1.0 + \left|\frac{\theta_f(\mathbf{c}) - \theta_i\big(S(\mathbf{c})\big)}{\pi}\right| * -2.0, \tag{10}$$

where $-1 \le \gamma(\theta_f(\mathbf{c})) \le 1$, in which $\gamma(\theta_f(\mathbf{c})) = 1$ means that $\theta_f(\mathbf{c})$ and $\theta_i\big(S(\mathbf{c})\big)$ are pointing at the same direction, and $\gamma(\theta_f(\mathbf{c})) = -1$ that they are pointing to opposite directions.

Now, the Growing Direction factor of a cell $\mathbf{c}$, $\varphi_g(\mathbf{c})$, is defined as

$$\varphi_g(\mathbf{c}) = \frac{\zeta\big(S(\mathbf{c})\big) * \gamma\big(\theta_f(\mathbf{c})\big) + 1.0}{2.0}, \tag{11}$$

where $-1 \le \varphi_g(\mathbf{c}) \le 1$, in which $\varphi_g(\mathbf{c}) = -1$ means that is less likely to reach the goal-door given how the door signs are set in $S(\mathbf{c})$, whereas $\varphi_g(\mathbf{c}) = 1$ means that is high likely. When $\varphi_g(\mathbf{c}) = 0$, it means that the Growing Direction factor is not sure about either the growing direction angle, or about the smaller and larger numbers. Hence, it can not indicate whether $\mathbf{c}$ is a very likely frontier.

4.3 Parity Factor

This factor considers the characteristics of a door sign to be either *even* or *odd*, the kind of information that is not explicitly available in the environment, but it can be easily inferred after the number recognition. The idea is to attribute a high probability to corridors that contain mostly door signs with the same parity than the goal-door. It is important to mention that this factor also considers the case in which a corridor contains both even and odd door signs since the probability is proportional to their respective amount.

To calculate the Parity factor, first the amount of door signs from $S(\mathbf{c})$ that has the same or different parity than the goal-door are counted, given by the functions $L^{=}\big(S(\mathbf{c})\big)$ and $L^{\neq}\big(S(\mathbf{c})\big)$, respectively. Then, for a cell $\mathbf{c} \in \mathbf{C}$, its Parity factor, $\varphi_p(\mathbf{c})$, is given by

$$\varphi_p(\mathbf{c}) = 0.5 + \frac{L^{=}\big(S(\mathbf{c})\big) - L^{\neq}\big(S(\mathbf{c})\big)}{\max\Big(L^{=}\big(S(\mathbf{c})\big) + L^{\neq}\big(S(\mathbf{c})\big), w_p\Big)} * 0.5 \tag{12}$$

where $0 \le \varphi_p(\mathbf{c}) \le 1$, and w_p is a threshold used to control the minimum amount of detected door signs that are necessary to this equation reaches 0 or 1. When $\varphi_p(\mathbf{c}) = 1$, it means that all the observed doors have the same parity than the goal-door, whereas $\varphi_p(\mathbf{c}) = 0$ is the opposite. When $\varphi_p(\mathbf{c}) = 0.5$, it means that $L^{\neq}$ and $L^{=}$ are equal, and therefore is not possible to ensure the parity of the segment $S(\mathbf{c})$.

4.4 Robot and Door Orientation Factors

The robot moves through the environment, and it detects door signs as they are in its path. Usually, the position of doors follows a pattern, that includes the possibility of existing doors only on horizontal or vertical corridors, for instance. Therefore, aiming to find the goal-door quickly,

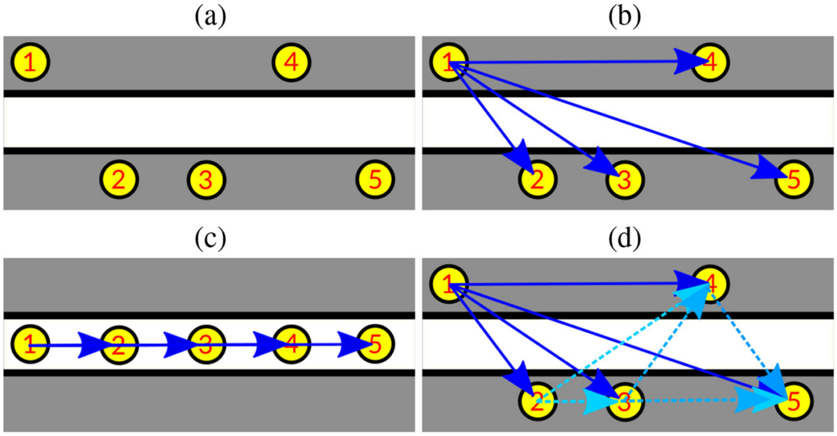


Fig. 3 Demonstration of how the increasing angle $\theta_i\left(S(\mathbf{c})\right)$ is computed in a segment. All the detected door signs within the segment, (**a**), are considered to calculate the $\theta_i\left(S(\mathbf{c})\right)$. The first step, (**b**), illustrates the vectors from door sign 1 to the other door signs, and it is easier to understand the effect of this vector calculation aligning all the door signs, (**c**). The final step of the vector computation, (**d**), shows all the vectors

it is better to prioritise corridors that are in the same orientation than the already visited ones containing many doors. If the robot can prioritise the corridors in the same orientation, by consequence, its most common orientation will be an angle similar to these corridors.

The scenario in Fig. 5 illustrates the importance of both Robot and Door Orientation factors. The robot starts at the intersection of corridors 1 and 2 and goes to the right, where it finds a second intersection between corridors 3 and 4, and it has to decide which one it should take. At this moment, all the door signs were detected while the robot had its orientation near 0°. Besides, the robot has most of the time moved heading 0°, as illustrated by the pink line in Fig. 5a. Hence, when the robot has to decide between corridor 3 or 4, both Robot and Door Orientation factors would guide the robot to corridor 3. It is due to the orientation of corridor 3 (∼360°), that has a smaller difference to 0° than to the orientation of corridor 4 (∼270°).

To calculate the Door Orientation factor, it is considered a history of the λ_d most recent robot's orientations when a door sign was detected. Based on this history, it is computed a histogram of such orientations, in which each bin saves the percentage of each possible robot's orientation. Then, given the orientation of $\mathbf{c}$, $\theta_f(\mathbf{c})$, it is consulted in the robot's orientation histogram the probability of finding a door sign considering such orientation,

$$\varphi_o(\mathbf{c}) = H_d[\theta_f(\mathbf{c})], \tag{13}$$

where $H_d[\cdot]$ is the door orientation histogram, and the Door Orientation factor is $0 \leq \varphi_o(\mathbf{c}) \leq 1$, in which 1 is 100% and 0 is 0%.

The Door and Robot Orientation factors are very similar to each other. The difference between them is that the first one saves the robot's orientation only when a door sign has been recognised. Therefore, it prioritises the $\theta_f(\mathbf{c})$ that has the highest $H_d[\theta_f(\mathbf{c})]$, i.e. the orientation in which the robot has detected most of the door signs. On the other hand, the idea of the second one, Robot Orientation factor, is to prioritise the $\theta_f(\mathbf{c})$ that is most similar to the robot's orientation that is more frequent, without considering when the door signs were recognised. This factor makes the robot takes into consideration other paths that despite not having door signs, may connect to other ones more promising.

As the robot moves through the environment, its λ_r most recent orientations are saved, and they are used to compute a histogram. Each histogram bin represents an angle and the percentage of it in the history of the robot's orientation. Given the calculated histogram, the $\theta_f(\mathbf{c})$ is used as an index to get the probability of that angle, as presented by

$$\varphi_r(\mathbf{c}) = H_r[\theta_f(\mathbf{c})], \tag{14}$$

where $H_r[\cdot]$ is the robot orientation histogram, and the Robot Orientation factor is $0 \leq \varphi_r(\mathbf{c}) \leq 1$, in which $\varphi_r(\mathbf{c}) = 1$ means that $\theta_f(\mathbf{c})$ is an orientation that is equal to the unique robot's orientation saved, whereas $\varphi_r(\mathbf{c}) = 0$ means that $\theta_f(\mathbf{c})$ is an orientation that the robot did not do.

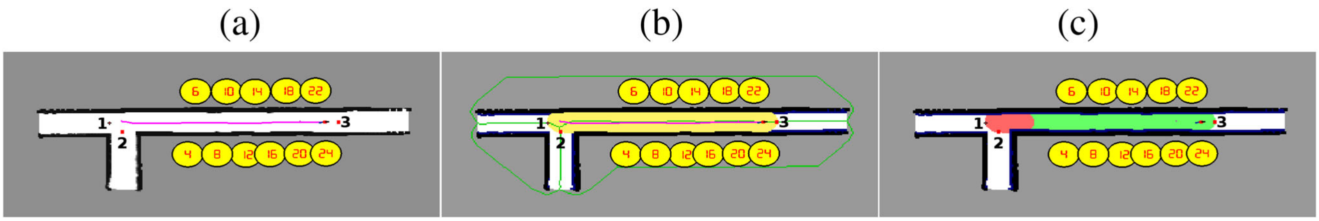


Fig. 4 Partial 2D map of the environment, showing three different corridors and many door signs. All images represent the same part of the environment, but (**a**) shows the simple 2D grid map, (**b**) shows the visited region, and (**c**) shows the two segments of the map

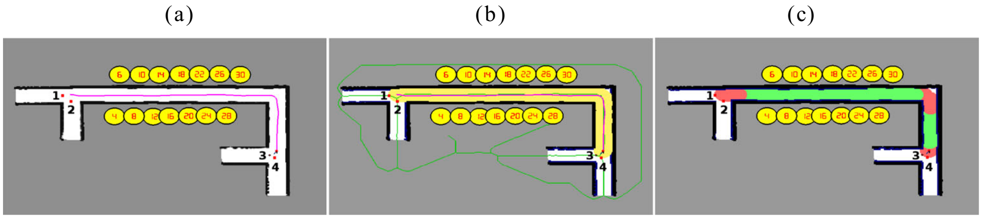


Fig. 5 Partial 2D map of the environment, showing four different corridors and many door signs. All images represent the same part of the environment, but (**a**) shows the simple 2D grid map, (**b**) shows the visited region, and (**c**) shows the five segments of the map

4.5 Distance Factor

The fifth factor that is considered by our Semantic Planner is the distance between the robot cell $\mathbf{c}_r$ and each $\mathbf{c} \in \mathbf{C}$, i.e. the smallest number of Voronoi cells that connects each pair of $(\mathbf{c}_r, \mathbf{c})$. Its goal is to guide the robot towards the closest $\mathbf{c}$, instead of spending battery and time going to a farthest one. Take the Fig. 6 as an example, and suppose that the goal-door is 71. The robot has started at the intersection between corridors 1 and 2, and it has moved to the right corridor, Fig. 6a. After a few minutes, guided by the Robot and Door Orientation factor, it has chosen to continue the searching on corridor 3. Even though this corridor has the same parity than the goal-door (both are odd) the Growing factor indicates that corridor 3 is not promising. Therefore, the robot should continue the searching in one of the other three options, corridors 1, 2 or 4. The Distance factor is responsible for indicating the closest option to the robot, given by the sum of green cells that connect the robot's current position and each red point near the numbers 1, 2 and 4, as shown in Fig. 6b.

The first step of the Distance factor calculation is to find the smallest distance $d_{\ll}$ between $\mathbf{c}_r$ and all $\mathbf{c} \in \mathbf{C}$, considering only the Voronoi cells in $\mathbf{M}$, in which one cell is considered as one to the distance sum. It is given by

$$\mathbf{c}_{\ll} = \arg\min_{\mathbf{c}\in\mathbf{C}} \big(D(\mathbf{c}_r, \mathbf{c})\big) \qquad (15)$$

where $D(\cdot, \cdot)$ is the function that counts the number of cells between two other specific cells. In this factor, only Voronoi cells are counted, regardless they are within mapped or unknown regions.

The idea is that the Distance factor of $\mathbf{c}$, $\varphi_d(\mathbf{c})$, should be high to small distances, and low to the big ones, i.e. give more preference to $\mathbf{c}$ that are closer to the robot. Then

$$\varphi_d(\mathbf{c}) = 1.0 - \left(1.0 - \frac{D(\mathbf{c}_r, \mathbf{c}_{\ll})}{D(\mathbf{c}_r, \mathbf{c})}\right)^4, \qquad (16)$$

where $0 \leq \varphi_d(\mathbf{c}) \leq 1$, in which $\varphi_d(\mathbf{c}) = 1$ means that $D(\mathbf{c}_r, \mathbf{c})$ is equal to $D(\mathbf{c}_r, \mathbf{c}_{\ll})$, and $\varphi_d(\mathbf{c}) = 0$ means that $D(\mathbf{c}_r, \mathbf{c})$ is so high that makes the division be around zero.

5 Experiments and Results

This section presents the results of simulated and physical experiments. Section 5.1 explains the software setup used in our simulated experiments, as well as the differences between the physical and simulated experiments. Section 5.2 presents the results of the simulation phase, comparing the performance of our proposed semantic AVS system and an entirely geometric AVS system, called *Greedy*. Four different maps were considered in this comparison. Section 5.3 introduces a second type of comparison, in which these two initial AVS systems, ours and the Greedy one, are compared to human participants teleoperating the robot in the simulation setup while performing the searching task. Finally, Section 5.4 demonstrates how our semantic AVS system performs in the physical world, as well as the information about where this test was performed.

It is also important to report the parameters used by our approach throughout all the experiments presented below, either in simulation or in the real world. Both w_g and w_p are set to eight. This means that in Eqs. 9 and 12, respectively,

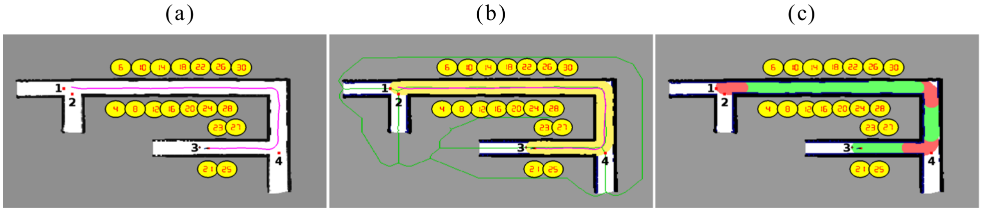


Fig. 6 Partial 2D map of the environment, showing four different corridors and many door signs. All images represent the same part of the environment, but (**a**) shows the simple 2D grid map, (**b**) shows the visited region, and (**c**) shows the six segments of the map

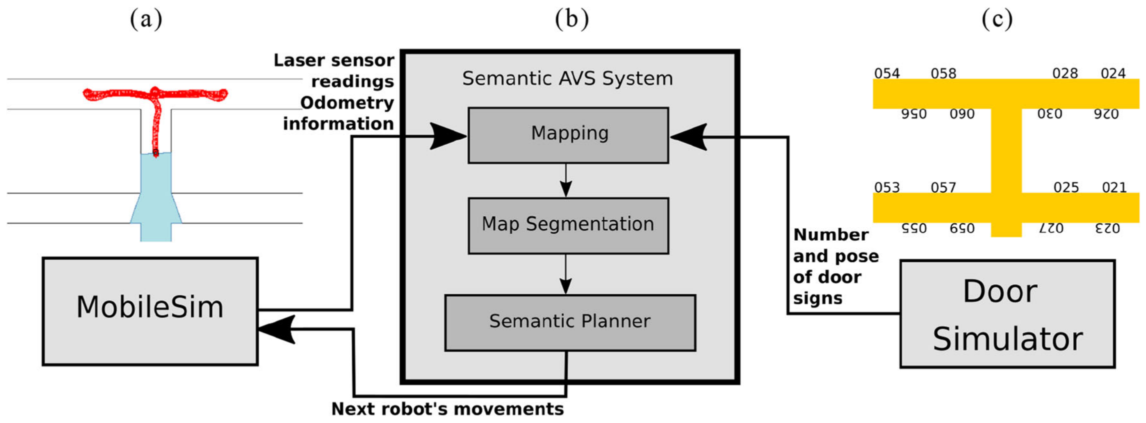


Fig. 7 Software setup used in the simulated experiments. It shows the MobileSim in (**a**), (**b**) represents the robot's and door signs information as input to our semantic AVS system that returns the robot's next movements, and (**c**) is the door signs map as ground-truth in the Door simulator. Both (**a**) and (**c**) represent the same position on the map

the closer or higher to eight the number of detected door signs is, more important the Growing Direction and Parity factors become. The number eight was chosen to balance the importance of the factors since a small number would make them important very soon in the search process, and a large number would play the opposite role. In addition to these two parameters, the Robot and Door orientation factor also have some parameters. The size of the $H_d[\cdot]$ is four, which is the outcome of dividing the range of [0°,179°] by 45°. It means our approach considers the robot's orientations when detecting a door sign in groups of 45° (e.g. if the robot detects a door sign and its orientation is 42°, $H_d[0]$ is incremented). For the histogram $H_d[\cdot]$, we consider the past 6.000 orientations, as we read hundreds of robot's orientation per minute, and this reading is noise. For the case of $H_r[\cdot]$, we assume a finer setup, since the robot may be in a different orientation in the range of [0°,359°]. The size of $H_r[\cdot]$ is 18, and we consider the past 600.000 readings, due to our high reading rate from the robot's orientation, the presence of noise in the data, and to reduce the impact of an unexpected turning that may happen.

5.1 Simulated Experiment Setup

The setup of the simulated experiments is represented in Fig. 7. The MobileSim simulates a Pioneer 3-DX robot equipped with a 180° Laser, providing its odometry information and its laser sensor readings, Fig. 7a. However, MobileSim does not provide information from door signs, which is vital for the tests in our paper. Therefore, we developed a door simulator (DS) to mimic both the two RGB cameras that are embedded in the physical robot and the Image Processing module that recognises the numbers, Fig. 7c. DS provides numbers of door signs and their positions in the world when they are within the robot's field of view. Then, the final setup is a combination of the MobileSim to read the robot's information, and our DS that provides the door signs information, as illustrated in Fig. 7.

The first evaluation of our semantic AVS system was made through the comparison with the Greedy AVS system in simulated indoor environments. The experiments considered four different scenarios, the Table 1 and the Fig. 8 present the details of them. The four scenarios vary considerably regarding the amount of door signs and how they are set, the size of the buildings, and the corridors orientation. The *Normal* and *Inverse* were made aiming to test the AVS systems in scenarios with many long corridors intersecting each other, where the AVS systems are forced to make decisions very often. Due to the high amount of door signs in both scenarios, four door signs were chosen as goal-doors for the tests, one in each horizontal corridor. Their difference is that *Normal*, Fig. 8a, has its door signs sequence increasing from the middle to the borders, whereas the *Inverse*, Fig. 8b, is in the other way around. This way, we can test the performance of our semantic AVS system in different door signs arrangement. The *Hotel* is the third scenario used in the experiments, and it is the third and fourth floors of the Hotel Pennsylvania [21] located in New York. With the highest amount of door signs and a large environment containing many door signs and long corridors, Fig. 8c, the *Hotel* scenario aims to test the AVS systems in terms of how our semantic AVS system analysis the numbers from door signs. A bad choice in *Hotel* may cause a long run that will not lead to the goal-doors. The fourth scenario is from a public dataset called KTH Campus,[1] Fig. 8d, that contains more than 38,000 rooms in total, considering the many floor plans from different

[1] It was used the left building from the floor plan identified as 0510028829_A30–00–07, A0043015. The dataset can be found at http://www.csc.kth.se/~aydemir/KTH_CampusValhallavagen_Floorplan_Dataset.tar.bz2

Table 1 Different scenarios used on our simulated tests

Name	# of Door signs	Goal-doors
Normal	113	54, 55, 111, 124
Inverse	116	54, 55, 111, 124
Hotel	124	76, 135, 148, 185
KTH	47	756

buildings [4]. Even though the particular floor plan chosen for this test, called *KTH* scenario, has the lowest amount of door signs compared to the other scenarios used in the tests, it presents corridors in a different orientation than the first three ones. All tests in the simulation were carried out in a laptop 8GB RAM and processor *i7*.

5.2 Semantic and Greedy AVS Systems

Our semantic AVS system was early introduced in Sections 3 and 4. In this section, the performance of our system is compared to the Greedy AVS system, which has the exact same basis presented in Section 3, but its planner is composed only by the geometric factor from Eq. 7, i.e. the Eq. 7 with $\alpha = 0$. Therefore, the planner of the systems is the only difference, that is responsible for the reasoning over their inputs. In summary, the Greedy AVS system searches for goal-doors based on the nearest frontier, whereas our semantic AVS system considers environmental information aiming to make smarter decisions.

Both systems were tested using the same simulation setup. For the four scenarios, the door signs shown in Table 1 were set as goal-doors. They were chosen to cover as many corridors of the scenarios as possible. For each goal-door, both systems repeated their respective tests ten times to have statistically significant results. For every test, it was measured, in meters, the distance travelled by the robot, from its initial position until it finds the goal-door. The distance travelled is the search cost used in our paper, and hence, the shorter the distance, the better is the system performance. Even though we have not measured and presented the search cost in terms of time, it is important to mention that both our Semantic AVS system and the Greedy one moved the robot with the same velocity. Therefore, the system that provides the shortest distance travelled is also the fastest system. Throughout the tests presented in this Section, in Eq. 1 the $a = 1$, and in Eq. 4 the $\delta = 2$. These parameters means that each cell of the 2D grid map $\mathbf{M}$ had value 1 to compute the kernel area and that the environment had only two types, corridor and not a corridor, as shown in Fig. 1c.

Tables 2, 3, 4, and 5 present the results of both approaches in each scenario in the simulated tests, *Normal*, *Inverse*, *Hotel*, and *KTH*, respectively. The colourful columns represent the

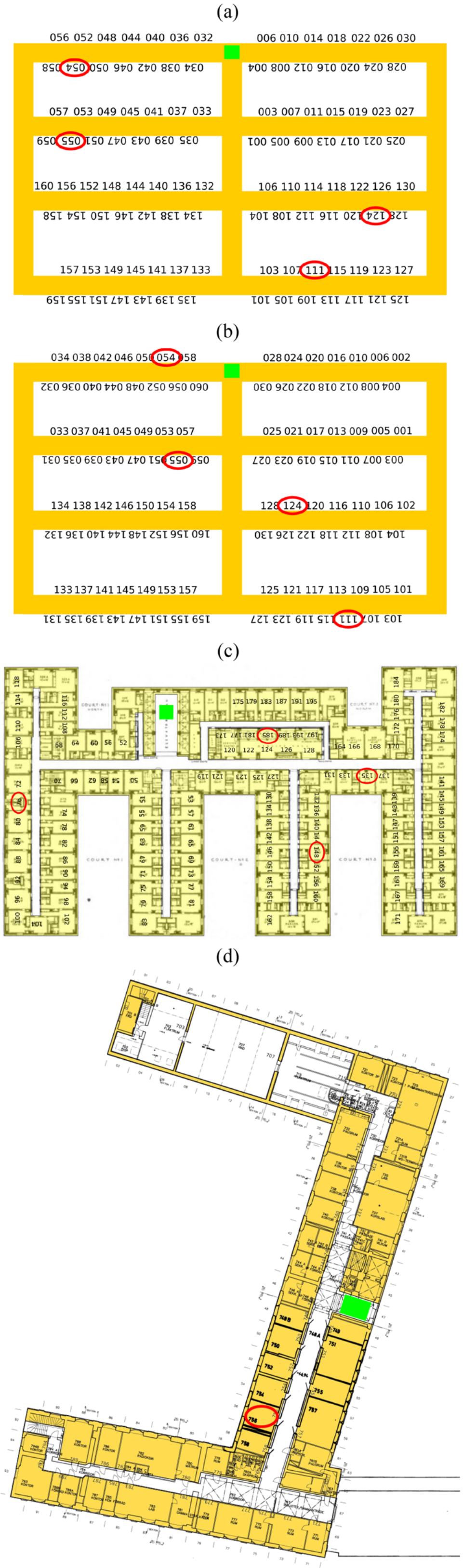
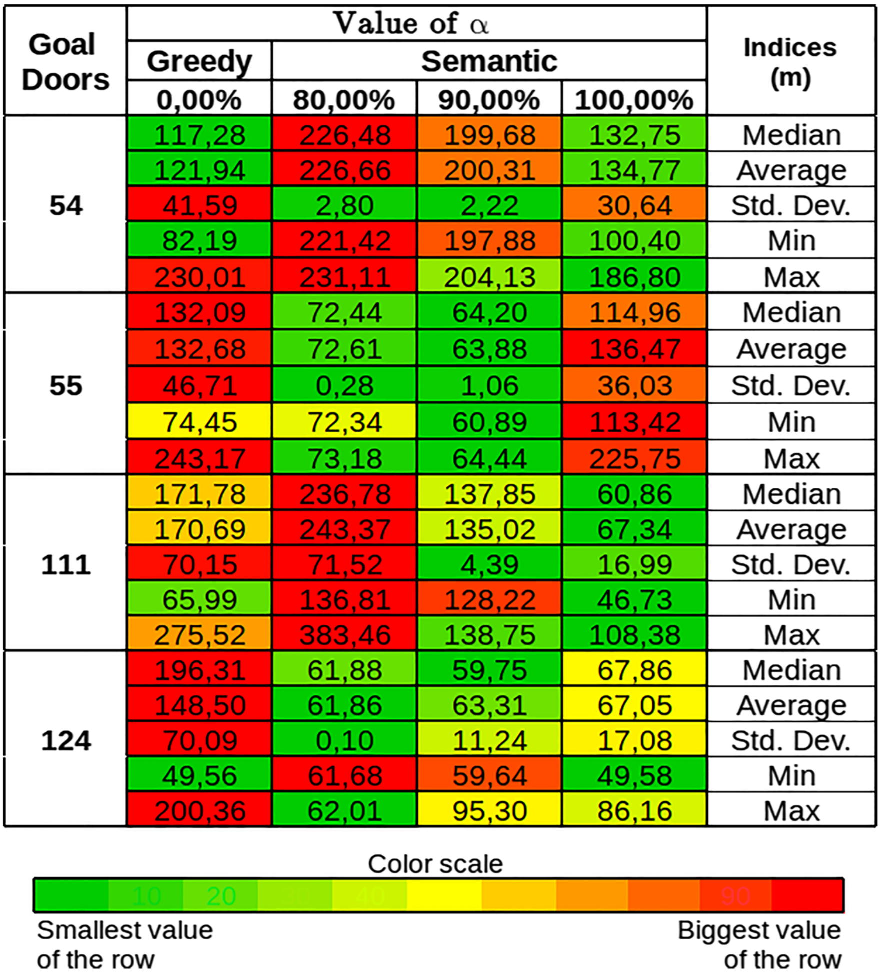


Fig. 8 The four maps used in the simulated experiments. The green squares represent the position where the robot has started, and the red circles highlight the goal-doors. The maps are *Normal* (**a**), *Inverse* (**b**), *Hotel* (**c**), and *KTH* (**d**)

Table 2 Results of the greedy and our semantic AVS systems in the *Normal* scenario. All the results are shown in meters



Goal Doors	Value of α				Indices (m)
	Greedy	Semantic			
	0,00%	80,00%	90,00%	100,00%	
54	117,28	226,48	199,68	132,75	Median
	121,94	226,66	200,31	134,77	Average
	41,59	2,80	2,22	30,64	Std. Dev.
	82,19	221,42	197,88	100,40	Min
	230,01	231,11	204,13	186,80	Max
55	132,09	72,44	64,20	114,96	Median
	132,68	72,61	63,88	136,47	Average
	46,71	0,28	1,06	36,03	Std. Dev.
	74,45	72,34	60,89	113,42	Min
	243,17	73,18	64,44	225,75	Max
111	171,78	236,78	137,85	60,86	Median
	170,69	243,37	135,02	67,34	Average
	70,15	71,52	4,39	16,99	Std. Dev.
	65,99	136,81	128,22	46,73	Min
	275,52	383,46	138,75	108,38	Max
124	196,31	61,88	59,75	67,86	Median
	148,50	61,86	63,31	67,05	Average
	70,09	0,10	11,24	17,08	Std. Dev.
	49,56	61,68	59,64	49,58	Min
	200,36	62,01	95,30	86,16	Max

Color scale

Smallest value of the row

Biggest value of the row

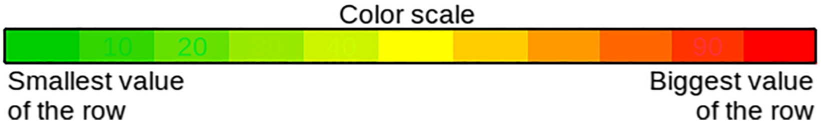
results achieved by tested AVS systems, in which the first is the result from Greedy AVS system, and the other three are from our semantic one. In the greedy column, the value 0, 00% means that $\alpha = 0\%$ in Eq. 7, and hence, the planner becomes fully geometric. In the semantic columns, the same value ranges from 80, 0% until 100, 0%, which means that the α ranges from 0.80 until 1.0 in Eq. 7. Hence, it changes the importance of the semantic factor in that equation. Our semantic AVS system was also tested with α ranging from 0.5 up to 0.7, but the results were not significant, and they are not presented in the tables. The rows of the tables correspond to the goal-doors used as targets, and each goal-door is evaluated in terms of Median, Average, Standard Deviation, Minimum and Maximum distances. It is also important to highlight that within a row, the colour of the table cells ranges from green to red. Green represents the cell with the smallest value within a row, and red represents the largest one.

The results in Table 2, *Normal* scenario, and in Table 3, *Inverse* scenario, are similar in terms of which column has the most red cells. In both cases, our semantic AVS system has a better performance in contrast to the greedy one, since most of the green cells are within the semantic columns, mainly when $\alpha = 80.0\%$ and $\alpha = 90.0\%$. It is also important to highlight that when there are green values within the greedy column, such as the case of door sign 54 in Table 2, its standard deviation is the highest one for that door sign, $41.59m$. In Table 3, the lowest average and minimum of the goal-door 111 are from the greedy AVS system, but again its standard deviation is the highest. It means that the ten tests of the greedy system vary considerably, as shown by the difference between its minimum and maximum values. On the other hand, the standard deviation within the semantic columns is lower, meaning that our semantic system has more constant behaviour during the search. It always guides the robot through the same path, making the same decisions in different executions.

The Tables 4 and 5, from *Hotel* and *KTH*, present similar results than the two previous tables, in terms of the greedy column having most of the red cells. Besides highlighting that, in general, our semantic AVS system has better performance than the greedy system, both tables also show that our proposed system is efficient in physical scenarios. Even though the $\alpha = 100.00\%$ column within

Table 3 Results of the greedy and our semantic AVS systems in the *Inverse* scenario. All the results are shown in meters

Goal Doors	Value of α				Indices (m)
	Greedy	Semantic			
	0,00%	80,00%	90,00%	100,00%	
54	82,03	29,97	25,22	131,57	Median
	98,88	29,68	24,98	135,18	Average
	41,52	0,89	0,90	15,13	Std. Dev.
	72,61	27,16	22,44	118,57	Min
	207,20	30,03	25,40	166,40	Max
55	152,67	69,26	41,35	128,48	Median
	154,29	62,34	46,28	109,80	Average
	54,78	10,38	8,13	41,57	Std. Dev.
	63,60	49,44	41,11	35,00	Min
	232,54	71,31	61,34	138,67	Max
111	273,64	250,07	219,97	214,02	Median
	200,16	230,27	205,45	209,96	Average
	102,57	54,63	45,38	14,65	Std. Dev.
	61,86	153,58	124,08	188,42	Min
	278,57	292,28	267,79	231,98	Max
124	205,26	177,16	160,49	173,34	Median
	165,73	145,59	137,05	156,20	Average
	82,65	42,73	38,69	60,25	Std. Dev.
	58,15	92,99	80,53	35,59	Min
	291,36	181,82	162,30	200,52	Max



the semantic columns presents satisfying results, mainly in Table 5, a purely semantic AVS system is not always suitable for searching tasks. The geometric factor in Eq. 7 is essential and combined with the semantic factor may provide the best results.

Besides the previous analysis, the optimal solution for each scenario was also measured. It is the shortest path between the starting position, green squares, and a goal-door, red circles, in Fig. 8. The Tables 6, 7, 8, and 9 present the optimal solution to each goal-door, as well as the average and standard deviation with each AVS system from Tables 2, 3, 4, and 5.

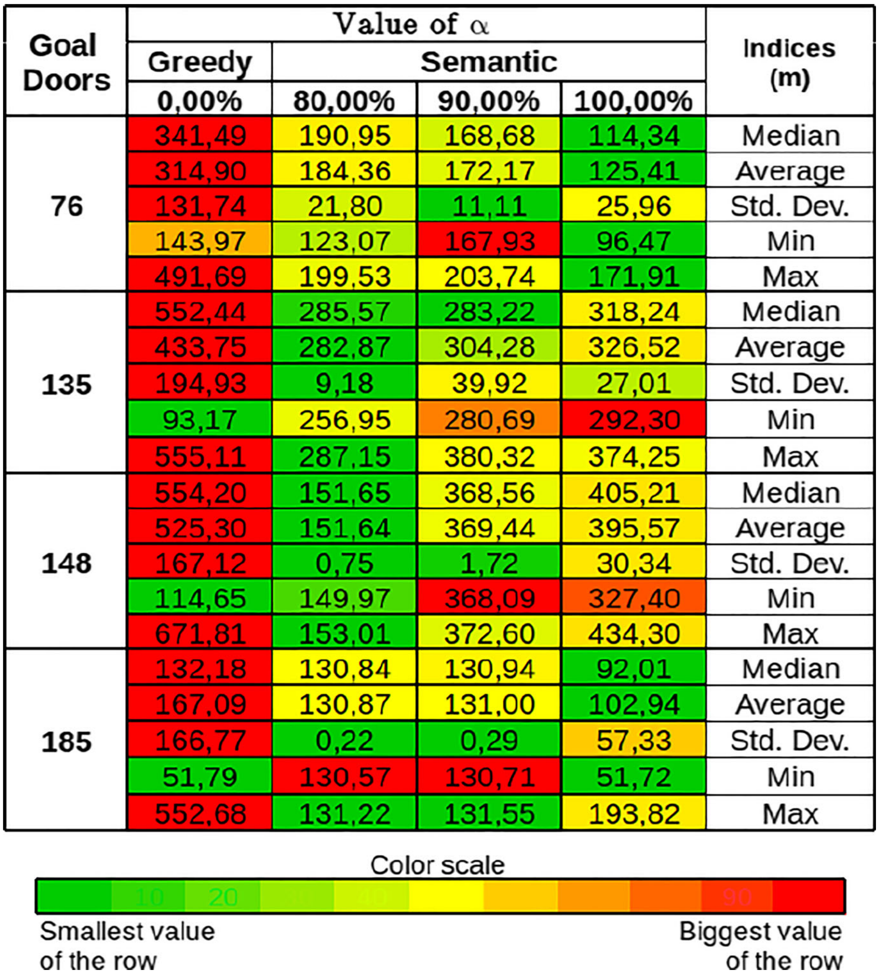
In general, the difference between the optimal solution of each goal-door and the averages in the same line is larger for results from greedy system than the ones from our semantic AVS system. For example, the optimal solution for the goal-door number 124 in Table 6 is 33.16m. The average of the Greedy system for the goal-door 124 is 148.5m ± 70.08m, which is almost five times larger than the shortest path. The results of our semantic AVS system for this same goal-door, in the worst case, is 67.04m ± 17.08m, which is just two times larger.

The same analysis can be made for other goal-doors in other scenarios, Tables 7 and 8. Besides this analysis, it is also possible to measure how large the averages are compared to their optimal solution. For all goal-doors and systems tested in the simulation experiments, Table 10 shows how many times, in percentage, the averages are larger compared to the optimal solutions. For the case of our semantic AVS system from Tables 6, 7, and 8, it is considered the lowest average between the ones from $\alpha = 80.0\%$, $\alpha = 90.0\%$, and $\alpha = 100.0\%$.

To compute the percentages presented in Table 10, it is considered the optimal solution of each goal-door for each scenario as 100%. Hence, if the average is larger than the shortest path, it will be higher than 100%, as the case of the goal-door 54, scenario *Normal*. For the greedy system, it is approximately seven times larger than the optimal solution, i.e. 709.39%.

In Table 10, most of the lowest percentages are within the semantic rows. There are few goal-doors in which the greedy system presents the lowest rate. That is the case of goal-door 54 of the *Normal* scenario, and the 111 of the *Inverse*. However, even though the greedy

Table 4 Results of the greedy and our semantic AVS systems in the *Hotel* scenario. All the results are shown in meters



Goal Doors	Value of α				Indices (m)
	Greedy	Semantic			
	0,00%	80,00%	90,00%	100,00%	
76	341,49	190,95	168,68	114,34	Median
	314,90	184,36	172,17	125,41	Average
	131,74	21,80	11,11	25,96	Std. Dev.
	143,97	123,07	167,93	96,47	Min
	491,69	199,53	203,74	171,91	Max
135	552,44	285,57	283,22	318,24	Median
	433,75	282,87	304,28	326,52	Average
	194,93	9,18	39,92	27,01	Std. Dev.
	93,17	256,95	280,69	292,30	Min
	555,11	287,15	380,32	374,25	Max
148	554,20	151,65	368,56	405,21	Median
	525,30	151,64	369,44	395,57	Average
	167,12	0,75	1,72	30,34	Std. Dev.
	114,65	149,97	368,09	327,40	Min
	671,81	153,01	372,60	434,30	Max
185	132,18	130,84	130,94	92,01	Median
	167,09	130,87	131,00	102,94	Average
	166,77	0,22	0,29	57,33	Std. Dev.
	51,79	130,57	130,71	51,72	Min
	552,68	131,22	131,55	193,82	Max

Color scale

Smallest value of the row

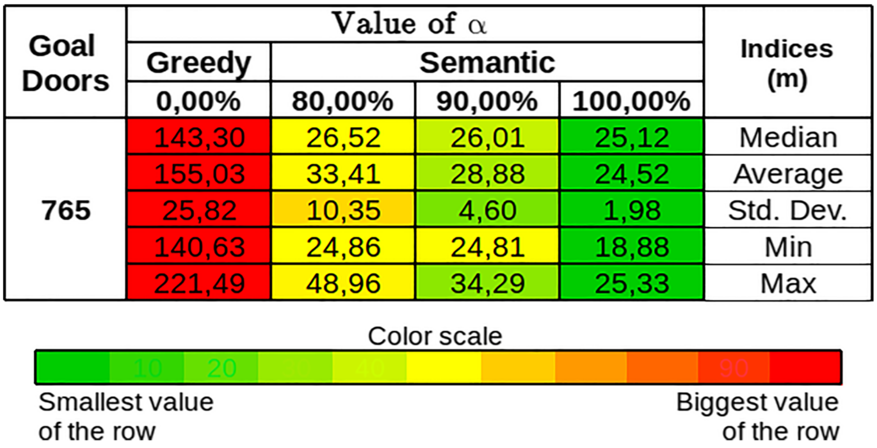
Biggest value of the row

system presents low values, the values from the semantic system to the same goal-doors are close. In contrast to this, analysing the goal-door 54 of the *Inverse* scenario, for instance, the value from the greedy system is almost four times larger.

5.3 Human Participants Performance in Object Searching Task

The results presented in Table 10 illustrate how many times, in percentage, the results of the greedy and our semantic

Table 5 Results of the greedy and our semantic AVS systems in the *KTH* scenario. All the results are shown in meters



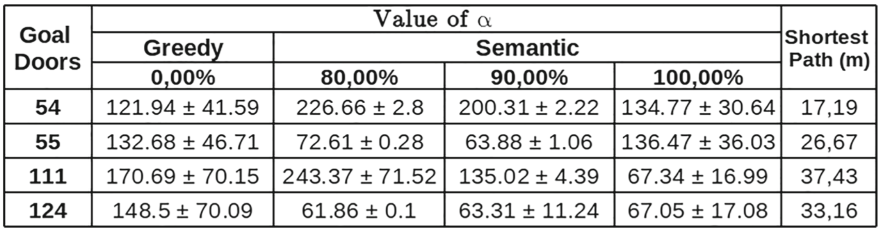
Goal Doors	Value of α				Indices (m)
	Greedy	Semantic			
	0,00%	80,00%	90,00%	100,00%	
765	143,30	26,52	26,01	25,12	Median
	155,03	33,41	28,88	24,52	Average
	25,82	10,35	4,60	1,98	Std. Dev.
	140,63	24,86	24,81	18,88	Min
	221,49	48,96	34,29	25,33	Max

Color scale

Smallest value of the row

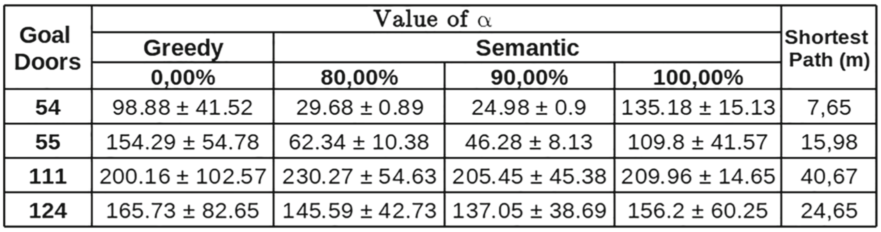
Biggest value of the row

Table 6 The average and the standard deviations from the *Normal* scenario, Table 2, and the optimal solution (shortest path) between each goal-door and the starting position. All the results are shown in meters



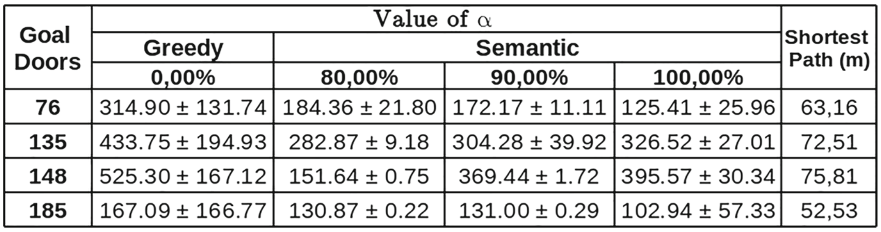
Goal Doors	Value of α				Shortest Path (m)
	Greedy	Semantic			
	0,00%	80,00%	90,00%	100,00%	
54	121.94 ± 41.59	226.66 ± 2.8	200.31 ± 2.22	134.77 ± 30.64	17,19
55	132.68 ± 46.71	72.61 ± 0.28	63.88 ± 1.06	136.47 ± 36.03	26,67
111	170.69 ± 70.15	243.37 ± 71.52	135.02 ± 4.39	67.34 ± 16.99	37,43
124	148.5 ± 70.09	61.86 ± 0.1	63.31 ± 11.24	67.05 ± 17.08	33,16

Table 7 The average and the standard deviations from the *Inverse* scenario, Table 3, and the optimal solution (shortest path) between each goal-door and the starting position. All the results are in meters



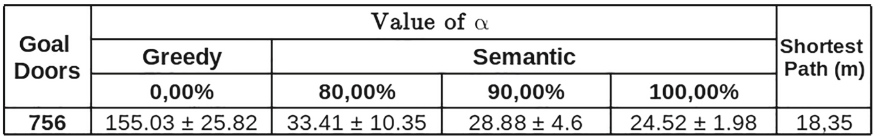
Goal Doors	Value of α				Shortest Path (m)
	Greedy	Semantic			
	0,00%	80,00%	90,00%	100,00%	
54	98.88 ± 41.52	29.68 ± 0.89	24.98 ± 0.9	135.18 ± 15.13	7,65
55	154.29 ± 54.78	62.34 ± 10.38	46.28 ± 8.13	109.8 ± 41.57	15,98
111	200.16 ± 102.57	230.27 ± 54.63	205.45 ± 45.38	209.96 ± 14.65	40,67
124	165.73 ± 82.65	145.59 ± 42.73	137.05 ± 38.69	156.2 ± 60.25	24,65

Table 8 The average and the standard deviations from the *Hotel* scenario, Table 4, and the optimal solution (shortest path) between each goal-door and the starting position. All the results are in meters



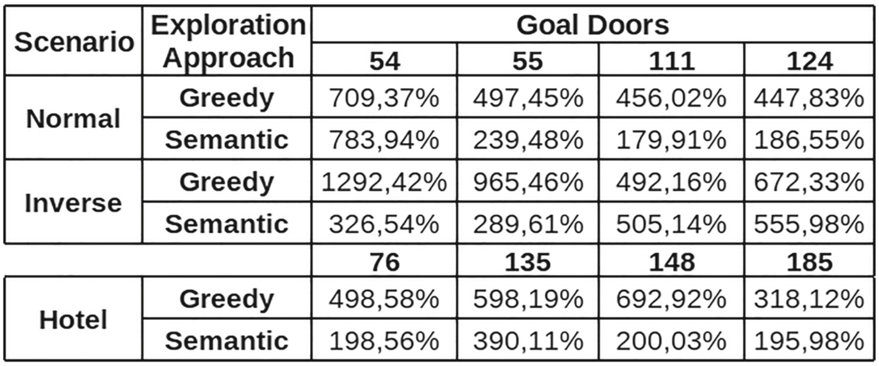
Goal Doors	Value of α				Shortest Path (m)
	Greedy	Semantic			
	0,00%	80,00%	90,00%	100,00%	
76	314.90 ± 131.74	184.36 ± 21.80	172.17 ± 11.11	125.41 ± 25.96	63,16
135	433.75 ± 194.93	282.87 ± 9.18	304.28 ± 39.92	326.52 ± 27.01	72,51
148	525.30 ± 167.12	151.64 ± 0.75	369.44 ± 1.72	395.57 ± 30.34	75,81
185	167.09 ± 166.77	130.87 ± 0.22	131.00 ± 0.29	102.94 ± 57.33	52,53

Table 9 The average and the standard deviations from the *KTH* scenario, Table 5, and the optimal solution (shortest path) between each goal-door and the starting position. All the results are in meters



Goal Doors	Value of α				Shortest Path (m)
	Greedy	Semantic			
	0,00%	80,00%	90,00%	100,00%	
756	155.03 ± 25.82	33.41 ± 10.35	28.88 ± 4.6	24.52 ± 1.98	18,35

Table 10 Comparison of the optimal solution (shortest path) of each goal-door from each scenario, with the averages from Tables 2, 3, and 4. The shortest path is equivalent to 100%, and the figure shows how large the averages are in comparison with the optimal solution



Scenario	Exploration Approach	Goal Doors			
		54	**55**	**111**	**124**
Normal	**Greedy**	709,37%	497,45%	456,02%	447,83%
	Semantic	783,94%	239,48%	179,91%	186,55%
Inverse	**Greedy**	1292,42%	965,46%	492,16%	672,33%
	Semantic	326,54%	289,61%	505,14%	555,98%
		76	**135**	**148**	**185**
Hotel	**Greedy**	498,58%	598,19%	692,92%	318,12%
	Semantic	198,56%	390,11%	200,03%	195,98%

AVS systems are larger than the optimal solution. Some results from the semantic system are two, three or even four times larger, whereas the greedy system provides results that are until 12 times larger than the optimal solution for the scenario *Inverse* and goal-door 54.

Given only these high percentages, it seems that both approaches are not suitable for the task of finding a target door sign in an unknown environment based on text information as visual cues. However, it is important to highlight that this task is challenging since the environment is unknown, and there is no way of planning an optimal path a priori. This section illustrates the difficulty level of the searching task by presenting an experiment in which human participants were invited to perform the searching while piloting the robot in the simulator presented in Section 5.1. For this experiment, it was measured (in meters) the distance travelled by the robot, from the initial position until the goal-doors. The distance travelled is the search cost used in our paper, and hence, the shorter the distance, the better is the system performance.

Instead of using the planner of either our semantic or the greedy AVS system to accomplish the finding the goal-door task, ten human participants were invited to teleoperate the robot in the simulation setup, to perform the same role than our semantic planner. The participants were presented to the searching task beforehand, with a time to get familiar with the robot control system and our simulation setup. This experiment aimed to measure human performance in the same setup as the other tested system, to show whether human reasoning provides better results than our semantic AVS system in the same conditions. Therefore, the humans controlled the robot in the same simulator and graphical interface as the two AVS systems, as shown by Fig. 1a. The difference of this experiment to the one from Section 5.2 is how the choices are made. In this case, the participants have to choose where the robot must go, playing the planner role to choose the path.

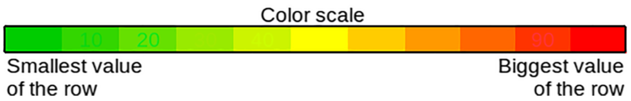
In this experiment, only two scenarios were tested, with one goal-door each. The *Normal* and *Inverse* scenarios are certainly similar, differing only on how the door signs are set. Given that humans are good at memorising what they have seen, it would be unfair to submit them to two similar scenarios, or more than one goal-door for the same scenario. Therefore, the *Normal* and *Hotel* scenarios were chosen. Door signs picked as the target were 111 to *Normal* and 148 to *Hotel*, because they are not too far nor too near to the initial position. Hence, the participants would have to explore at least a small part of both scenarios. Throughout all the tests, the only data considered to the evaluation was the travelled distance.

Table 11 summarises the analysis of the ten participants. Besides, it also compares human performance to the greedy and our semantic AVS systems. As in the previous tables, green represents the cells with the smallest value within a row, and red represents the largest one. As can be seen, our semantic AVS system presents a smaller average in both goal-doors, with the lowest indices compared to the others. The minimum travelled distance for the goal-door 148 is the only case in which our semantic system does not have the lowest result.

In contrast to our semantic system, the greedy one presents the worst results for both goal-doors, which confirms the previous results presented in the Section 5.2. It is also worth to mention the high standard deviation of the Human participants for both goal-doors. It shows that they had different performances, in which some had a shorter travelled distance than others. Some participants did not follow the same pattern when making decisions throughout their run. At the end of their participation, they have reported that they did not have an efficient strategy to search for the target, and no reasoning was made based on the door signs. One specific participant temporarily forgot the goal-door for the *Hotel* scenario, even though it was written in a paper that was in front of the participants. The

Table 11 Human performance to the problem of AVS system. It is compared to the greedy and semantic systems, in which all of them had to find two goal-doors, one in each scenario. The results are presented in meters

		Different approaches			
Scenario	Goal Doors	Humans	Greedy	Semantic	Indices (m)
Normal	111	109,06	171,78	60,86	Median
		103,38	170,69	67,34	Average
		34,00	70,15	16,99	Std. Dev.
		63,88	65,99	46,73	Min
		170,94	275,52	108,38	Max
Hotel	148	261,17	554,20	151,65	Median
		299,30	525,30	151,64	Average
		136,10	167,12	0,75	Std. Dev.
		75,23	114,65	149,97	Min
		531,96	671,81	153,01	Max



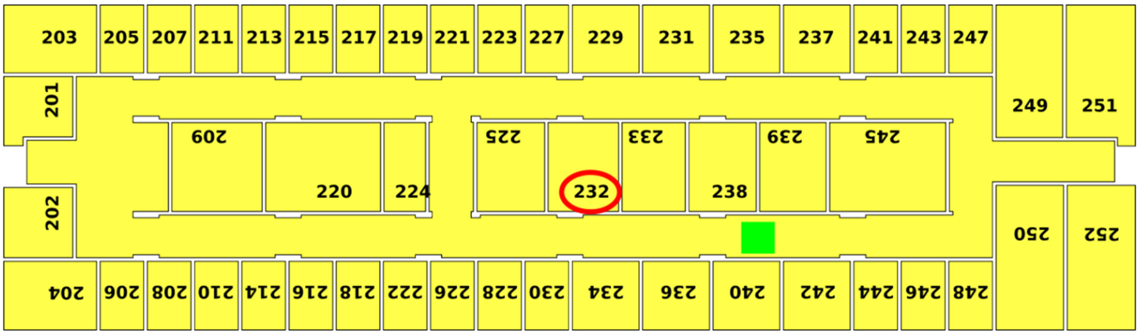


Fig. 9 Map used in the experiments with the real robot. It is the building where the Phi Robotics Research Lab is located at the Federal University of Rio Grande do Sul, Brazil. The green square represents the position where the robot starts, and the red circle highlight the goal-door

human eye has a wider field of view than the two cameras used in this experiment setup, which provides advantages to humans when searching for objects in an unknown environment. However, as previously mentioned, the goal of this experiment was to test humans as the decision-maker within the system. That is why they used the same software setup for the experiments as our semantic and greedy systems.

Besides the lowest average from our semantic AVS system, there are other advantages in comparison to the participants' results. Its small standard deviation means that the same decisions were taken in all the ten test repetitions, which suggests that our system is not random when it is being executed. On the other hand, the same does not apply for the Human results, which means that every participant has their particular reasoning to make a decision. Hence, some participants are more efficient than others in this kind of tasks. Besides the standard deviation, another advantage is the fact that robots are not disturbed by other moving objects or agents in the environment. Therefore, they can focus on the task, and they do not forget the goal-door, what happened to one of the humans during the experiment.

5.4 Experiments Using a Physical Robot

The experiment with a physical robot was performed in one of the buildings of the Federal University of Rio Grande do Sul, Brazil, where the Phi Robotics Research Lab is located. Figure 9 shows how this building is organised, as well as the rooms and their door signs. The robot used in this experiment is a Pioneer 3DX from MobileRobots, which is equipped with a Lidar laser scan of 180° and two RGB cameras, as shown in Fig. 10.

The goal of this experiment is to prove that our semantic AVS system works in physical scenarios, meaning that the robot should be able to find the target goal-door travelling the shortest distance as possible. For this experiment, the goal-door 232 is chosen as the target, which is located on the left side of the initial position (green square), Fig. 9. Even though it was at the same corridor as the initial position, the experiment setup is good to prove that our system can reason over the detected door signs. From the initial position, the robot can turn to the left or the right. If it decides the left direction, it will find the goal-door quicker, but the other direction would take it to the opposite side. For this case, as soon as a few door signs have been detected, our semantic system would be able to reason over them and infer that this direction is not promising, and hence, it should change to the opposite direction. Therefore, this would show that the door signs are used by our semantic system to find the goal-door efficiently.

The performance of our proposed semantic AVS system, when submitted to finding the goal-door 232 in a physical environment, was similar to the situation described above. Figure 11 depicts six steps of the system, and all of them

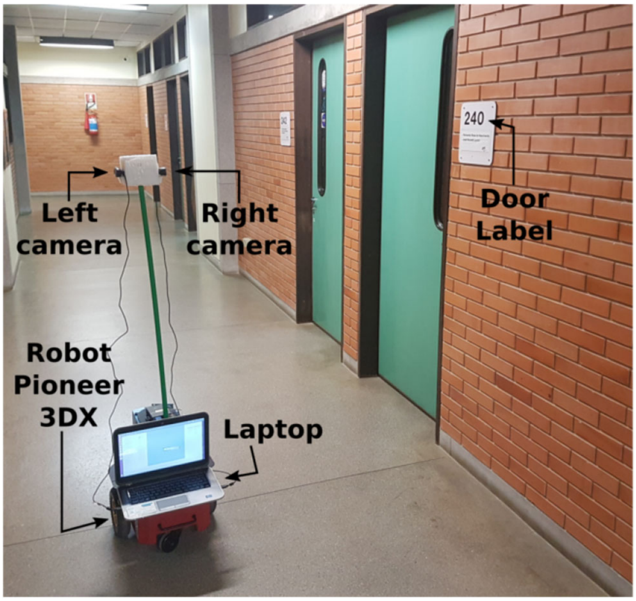


Fig. 10 The Pioneer 3DX robot used in the real environment experiment, as well as the two embedded cameras. The door signs of the environment are also depicted on this figure

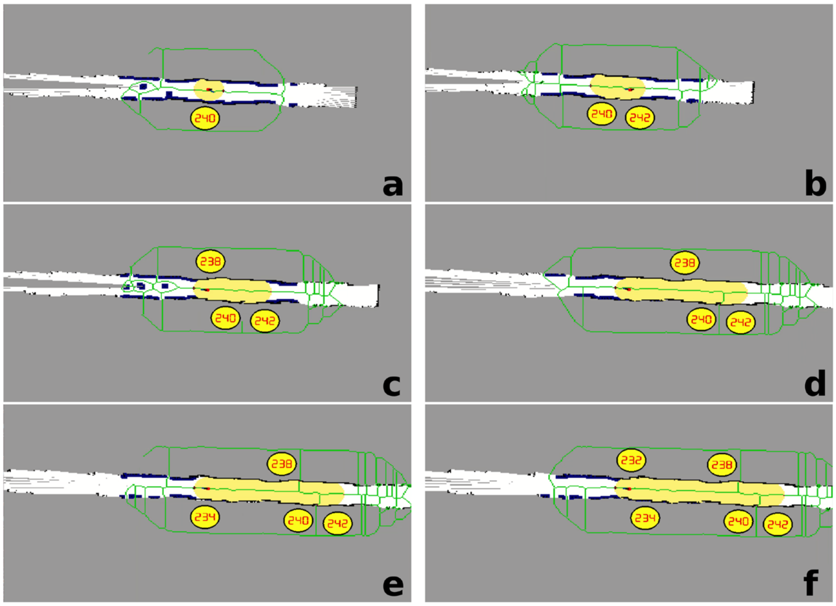


Fig. 11 Step-by-step of the performance of our semantic AVS system in the physical environment. *a* shows the initial position, where the robot has started the searching, whereas *f* shows the final step when the robot has found the goal-door 232

show important moments for the searching. In Fig. 11a, the robot just had finished one complete rotation to map its surroundings and detected the door sign 240. In Fig. 11b, the robot has chosen to turn right, where the door sign 242 has been found. This figure demonstrates how our planner decides on changing: *i*) the door signs 240 and 242 were recognised in an increasing sequence, and it means that as the robot goes forward, the distance from the goal-door just increases; *ii*) the robot is between two frontiers, and even though the robot is closer to the one that is in front of it, the other is not that far from it; *iii*) the first two door signs were found in a horizontal corridor, and so far the horizontal corridors are the more likely ones to contain other door signs. Combining all this information, the decision at this moment is that the frontier that the robot is following is less promising than the other one that is behind it. In Fig. 11c, the robot has changed its orientation to the opposite one. The door sign 238 was recognised, which supports the orientation changing. In Fig. 11d, as the robot has not recognised any other door sign that contradicts its decision, the exploration continues towards the left direction. In Fig. 11e, it recognises the door sign 234, which indicates a decreasing sequence towards the goal-door. Finally, in Fig. 11f, the goal-door 232 is detected, and the robot finishes the searching. Unfortunately, due to the COVID-19 pandemic situation, the university where the robot is located and this experiment was conducted is closed, as the health organisations recommend. Therefore, we were not allowed to perform more experiments like this to other goal-doors to show further the good performance of our semantic AVS system in the physical world.

6 Conclusion

We proposed a semantic AVS system that relies on semantic information inferred from text within the environment. The proposed system aims to demonstrate that it is possible to take advantage of different sources of information, such as door signs, traffic signs, or outdoor advertisement. Even though our paper has not tested all these different sources, only door signs, the results show that usage of text inferred from signs for robotic solutions is promising. Besides, our paper also intends to encourage the mobile robotics research community to explore the advantages of semantic information for mobile robot tasks. The main contributions of this paper are:

- a robust semantic planner, based on five different factors, that reason over the door signs to find the goal-door travelling the shortest possible distance;
- a semantic AVS system which, by using our semantic planner, can reason over the door signs and estimate when the robot is in a non-promising path without any training step;
- an analysis of the usage of text information as input to the semantic planner within the AVS system. In general, the analysis shows that our system has better results than humans participants, both in the same simulation setup.

Our semantic planner applied to the AVS system presented an excellent performance, mainly when compared to the results from the greedy system and humans performing the searching. Besides providing the shortest distance travelled, and by consequence, it was also the fastest search system, given that the robot moved with the same velocity in all the experiments. It is important to mention that no experiments were conducted in an environment in which its rooms were randomly signed. This is because we believe that in such kind of environment probably not even humans would be able to rely on the random signs and efficiently search for the target door sign, and our semantic AVS system would rely on an input that is not reliable. On the other hand, despite not being random, the *Hotel* map presents very challenging door sign configurations. Some corridors have a door sign sequence that does not increase nor decrease, and that does not have a predominant parity. These conditions do not reflect a well-structured environment, but our proposal still presented a robust performance, with better results than the other tested approach and humans.

Our semantic planner does not require that the door signs of the environment are set according to a specific pattern, as confirmed by the wide variety of the four tested scenarios. The scenario *Hotel* demonstrates how different the door signs can be located, and according to the feedback from the participants after their participation, the *Hotel* is indeed a little bit confusing for them.

The results show that our semantic AVS system presented better or similar results than the ones from human participants. However, it is important to highlight that these results were obtained when humans were piloting the robot in the same simulation setup as the other experiments. The human eyes have a wider field of view than cameras, so in this case, it would be unfair to compare the performance of humans against robots if they had different visual sensors. That is why only human reasoning was considered in this paper.

Finally, as future work, we aim to perform more experiments with the physical robot in real world to measure the performance of our proposal in different scenarios. We also intend to make our code publicly available, as it is implemented based on the Robot Operating System (ROS) and hence, it can be easily reused by the research community. The same applies to the door sign simulator developed by us for testing our proposal. About the type of the door sign, we also aim to explore other standards that also includes letters, not only numbers. This achievement would make our proposal suitable for applications in a wider range of environments. Besides, the potential of machine learning could be applied to the object searching problem. Instead of modelling the growth factor of a sequence or the odd and even factors, a machine learning-based system could learn how the door signs are set within the environment. In addition to that, investigating the other gains of textual information, such as reading the traffic signs to improve the driving performance of autonomous cars, is another topic that should be investigated. Lastly, our approach depends on the map segmentation, which is done by KDE, to group the detected door signs. In KDE, the kernel size changes the total amount and the area of the segments, besides being one of the parameters of our proposal. Therefore, the dependency of this parameter by our proposal should be investigated to make it more robust and stable.

Acknowledgments This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001, CNPq. Besides, this work is also partially supported by the Research Council of Norway (RCN) as a part of the COINMAC project (grant agreement 261645), the MECS project (grant agreement 247697) and the VIROS project (grant agreement 288285).

Author Contributions MM, DP, RM, EP and MK conceived and designed the approach. MM, and RM carried out the experiments, and the data analysis was performed by MM, DP, RM and MK. All authors wrote the manuscript and reviewed its final version.

Data Availability The code, data and any other document will be release in the repository of the Phi Robotics Research Lab,[2] as soon as all the files meet the Google Style Guide and are well documented.

Compliance with Ethical Standards

Consent for Publication The authors declare there is no conflict of interest in this paper.

Consent to Participate and to Publish Before carrying out the experiments, all participants who have collaborated in the experiments of this work have signed a consent term. It contains an explanation about the experiment, its goal, and the use of the data generated by them. The term also contains information about the confidentiality and the security of their participation.

References

1. Amherst, U.: Room numbering guidelines (2012)
2. Aydemir, A., Göbelbecker, M., Pronobis, A., Sjöö, K., Jensfelt, P.: Plan-based object search and exploration using semantic spatial knowledge in the real world. In: ECMR, pp. 13–18 (2011)
3. Aydemir, A., Jensfelt, P.: Exploiting and modeling local 3d structure for predicting object locations. In: International Conference on Intelligent Robots and Systems, pp. 3885–3892. IEEE (2012)
4. Aydemir, A., Jensfelt, P., Folkesson, J.: What can we learn from 38,000 rooms? reasoning about unexplored space in indoor environments. In: International Conference on Intelligent Robots and Systems, pp. 4675–4682. IEEE (2012)
5. Aydemir, A., Pronobis, A., Göbelbecker, M., Jensfelt, P.: Active visual object search in unknown environments using uncertain semantics. In: Transactions on Robotics, vol. 29, pp. 986–1002. IEEE (2013)
6. Aydemir, A., Pronobis, A., Sjöö, K., Göbelbecker, M., Jensfelt, P.: Object search guided by semantic spatial knowledge. In: The RSS, vol. 11 (2011)

[2] https://github.com/phir2-lab

7. Aydemir, A., Sjöö, K., Folkesson, J., Pronobis, A., Jensfelt, P.: Search in the real world: Active visual object search based on spatial relations. In: International Conference on Robotics and Automation, pp. 2818–2824. IEEE (2011)
8. Barber, R., Crespo, J., Gomez, C., Hernandez, A.C., Galli, M.: Mobile robot navigation in indoor environments: Geometric, topological, and semantic navigation IntechOpen (2018)
9. Begum, M., Karray, F.: Visual attention for robotic cognition: A survey. In: Transactions on Autonomous Mental Development, vol. 3, pp. 92–105. IEEE (2010)
10. Borenstein, J., Koren, Y.: Histogramic in-motion mapping for mobile robot obstacle avoidance. In: Transactions on Robotics and Automation (1991)
11. Borji, A., Cheng, M.M., Hou, Q., Jiang, H., Li, J.: Salient object detection: A survey. In: Computational visual media, pp. 1–34. Springer (2019)
12. Chen, S., Li, Y., Kwok, N.M.: Active vision in robotic systems: A survey of recent developments. In: The International Journal of Robotics Research, vol. 30, pp. 1343–1377 (2011)
13. Chung, T.H., Hollinger, G.A., Isler, V.: Search and pursuit-evasion in mobile robotics. In: Autonomous robots, vol. 31, p. 299. Springer (2011)
14. DiCarlo, J.J., Zoccolan, D., Rust, N.C.: How does the brain solve visual object recognition? In: Neuron, vol. 73, pp. 415–434. Elsevier (2012)
15. Ekvall, S., Kragic, D., Jensfelt, P.: Object detection and mapping for service robot tasks. In: Robotica, vol. 25, pp. 175–187 (2007)
16. Girdhar, Y., Whitney, D., Dudek, G.: Curiosity based exploration for learning terrain models. In: International Conference on Robotics and Automation. IEEE (2014)
17. Guo, Z., Hall, R.W.: Parallel thinning with two-subiteration algorithms. In: Communications of the ACM, vol. 32, pp. 359–373. ACM (1989)
18. Haines, B.: A basic model for numbering your rooms and spaces (2014)
19. Jung, K., Kim, K.I., Jain, A.K.: Text information extraction in images and video: A survey. In: Pattern Recognition (2004)
20. Maffei, R., Jorge, V.A.M., Rey, V.F., Franco, G.S., Giambastiani, M., Barbosa, J., Kolberg, M., Prestes, E.: Using n-grams of spatial densities to construct maps. In: International Conference on Intelligent Robots and Systems (2015)
21. McKim: Hotel - Pennsylvania typical floor plan. https://bit.ly/36Y6t5P (1919)
22. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: Conference on Computer Vision and Pattern Recognition (2012)
23. Prestes, E., Engel, P.M., Trevisan, M., Idiart, M.A.: Exploration method using harmonic functions. In: Robotics and Autonomous Systems (2002)
24. Quattrini Li, A., Cipolleschi, R., Giusto, M., Amigoni, F.: A semantically-informed multirobot system for exploration of relevant areas in search and rescue settings. In: Autonomous Robots (2016)
25. Rasouli, A., Lanillos, P., Cheng, G., Tsotsos, J.K.: Attention-based active visual search for mobile robots. In: Autonomous Robots, vol. 44, pp. 131–146. Springer (2020)
26. Rasouli, A., Tsotsos, J.K.: Integrating three mechanisms of visual attention for active visual search. ArXiv:1702.04292 (2017)
27. Rogers, J.G., Christensen, H.I.: Robot planning with a semantic map. In: International Conference on Robotics and Automation (2013)
28. Saidi, F., Stasse, O., Yokoi, K.: Active visual search by a humanoid robot. In: Robotics: Viable Robotic Service to Human, pp. 171–184. Springer (2007)
29. Schulz, R., Talbot, B., Lam, O., Dayoub, F., Corke, P., Upcroft, B., Wyeth, G.: Robot navigation using human cues: A robot navigation system for symbolic goal-directed exploration. In: International Conference on Robotics and Automation (2015)
30. Sjöö, K., Aydemir, A., Jensfelt, P.: Topological spatial relations for active visual search. In: Robotics and Autonomous Systems, vol. 60 (2012)
31. Sjöö, K., López, D.G., Paul, C., Jensfelt, P., Kragic, D.: Object search and localization for an indoor mobile robot. In: Journal of Computing and Information Technology, vol. 17. SRCE-University Computing Centre (2009)
32. Talbot, B., Lam, O., Schulz, R., Dayoub, F., Upcroft, B., Wyeth, G.: Find my office: Navigating real space from semantic descriptions. In: International Conference on Robotics and Automation (2016)
33. Tsotsos, J.K.: On the relative complexity of active vs. passive visual search. In: International journal of computer vision, vol. 7, pp. 127–141. Springer (1992)
34. University, S.: Room numbering guidelines (2017)
35. Veiga, T.S., Miraldo, P., Ventura, R., Lima, P.U.: Efficient object search for mobile robots in dynamic environments: Semantic map as an input for the decision maker. In: International Conference on Intelligent Robots and Systems (2016)
36. Ye, Y., Tsotsos, J.K.: On the collaborative object search team: A formulation. In: Distributed Artificial Intelligence Meets Machine Learning Learning in Multi-Agent Environments, pp. 94–116. Springer (1996)
37. Ye, Y., Tsotsos, J.K.: A complexity-level analysis of the sensor planning task for object search. In: Computational Intelligence, vol. 17, pp. 605–620 (2001)
38. Zeng, Z., Röfer, A., Jenkins, O.C.: Semantic linking maps for active visual object search. ArXiv:2006.10807 (2020)
39. Zhang, H., Zhao, K., Song, Y.Z., Guo, J.: Text extraction from natural scene image: A survey. In: Neurocomputing (2013)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mathias Mantelli received his B.Sc. in Computer Science from Federal University of Mato Grosso in 2014. He then went on to receive his M.Sc. in Computer Science from Federal University of Rio Grande do Sul in 2016 and since 2017, he has been a Ph.D. student at same university. He is member of the Phi Robotics Research Group and his research areas include localization, SLAM, and mobile robotics.

Diego Pittol received his B.Eng. degree in Computer Engineering from University of Santa Cruz do Sul (UNISC) in 2015. He then went on to receive his M.Sc degree in Computer Science from Federal University of Rio Grande do Sul (UFRGS) in 2018. His current research interests include mobile robotics, autonomous exploration, semantic SLAM and semantic exploration.

Renan Maffei received his M.Sc. and Ph.D. degrees in Computer Science from the Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, in 2013 and 2017, respectively. Currently he is Adjunct Professor at the UFRGS and member of the Phi Robotics Research Group. His research interests include localization, SLAM and integrated exploration with mobile robots.

Jim Torresen is a professor of computer science at the University of Oslo, Norway. His research interests include nature-inspired computing, adaptive systems, reconfigurable hardware, and robotics and their use in complex realworld applications. He received a PhD in computer science from the Norwegian University of Science and Technology. He is a senior member of the IEEE.

Edson Prestes is Professor at Institute of Informatics of the Federal University of Rio Grande do Sul, Brazil. He is leader of the Phi Robotics Research Group, past head of the Theoretical Informatics Department, co-head of the Intelligent Robotics and Computation Vision CNPq Group and CNPq Research Fellow. He received his B.Sc. in Computer Science from the Federal University of Para (1996), Brazil, and M.Sc. (1999) and Ph.D. (2003) in Computer Science from Federal University of Rio Grande do Sul. Edson is IEEE Senior Member and Member of the IEEE Robotics and Automation Society (IEEE RAS) and IEEE Standards Association (IEEE SA). Over the past years, he has been working in different international initiatives related to Standardisation, Humanitarian Activities, Artificial Intelligence, Robotics and Ethics. In these groups, Edson has been served in various roles as chair, vice-chair and member. For instance, he is chair of the IEEE RAS/SA P7007 — Ontological Standard for Ethically Driven Robotics and Automation Systems Working Group; vice-chair of the IEEE RAS/SA Ontologies for Robotics and Automation Working Group (ORA WG); founding chair of the IEEE South Brazil RAS Chapter; Fellow of the EP3 Foundation; member of The European AI Alliance; member of the Affective Computing and Embedding Values into Autonomous Intelligence Systems Committees at IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems; member of the IEEE RAS Special Interest Group on Humanitarian Technology and member of several IEEE RAS/SA Standardisation Working Groups.

Mariana Kolberg is a professor at Institute of Informatics of the Federal University of Rio Grande do Sul, Brazil. She is member of the Phi Robotics Research Group, member of the Intelligent Robotics and Computation Vision CNPq Group and CNPq Research Fellow. She received her B.Sc. (2002), M.Sc. (2005) and Ph.D (2009) in Computer Science from Pontifical Catholic University of Rio Grande do Sul. Mariana is member of IEEE and the IEEE Robotics and Automation Society (IEEE RAS). She was member of the IEEE RAS Special Interest Group on Humanitarian Technology and a founding Administrative Committee member of the IEEE South Brazil RAS Chapter. Over the past years, she has worked in different international initiatives related to Humanitarian Activities, Artificial Intelligence, Robotics. During her academic life, Mariana has served as reviewer for high impact International Journals and conferences and as reviewer for Brazilian Funding Agencies. Her research interests include robotics, interval robotics, mobile robotics and interval analysis.