Semantic Maps for Domestic Robots

João Miguel Camisão Soares de Goyri O'Neill

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. Rodrigo Martins de Matos Ventura
Prof. Pedro Daniel dos Santos Miraldo

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Rodrigo Martins de Matos Ventura
Member of the Committee: Prof. Plinio Moreno López

October 2015

Resumo

Dado o aumento de aplicações de robôs e particularmente de robôs de serviço, tem surgido na comunidade da Inteligência artificial a questão de como gerar comportamento *inteligente*. Embora se tenha até agora respondido a esta questão com modelos muito completos e rígidos do ambiente. Cada vez mais se aborda o problema com modelos mais simples e que podem aparentar ser incompletos mas que na verdade se controem à medida que interagem com o ambiente tornando-se progressivamente mais eficientes. Neste trabalho será apresentado um mapa semântico que terá o conhecimento fundamental para completar a tarefa de determinar a localização de objectos no mundo. Esta tarefa utiliza o módulo de reconhecimento de objectos para experienciar sensorialmente o ambiente, um planeador de acções e um mapa semântico que recebe informação de baixo nível do reconhecedor e a converte em informação de alto nível para o planeador. A sua architectura foi desenhada tendo em conta que é suposto que o mapa semântico seja utilizado por todos os módulos. Vários testes foram realizados em cenários realistas e utilizando objectos do dia a dia. As experiências mostram que o uso do mapa semântico torna o processo mais eficiente a partir da primeira interação com o ambiente.

Palavras-chave: Mapas semanticos, Procura de Objectos, Inferência Probabilística, Reconhecimento de Objectos

Abstract

Due to the increasing application of robots and particularly servicing robots, the question of how to generate *intelligent* behavior is progressively gaining importance in the Artificial Intelligence community. Although the solution to this issue was thought to be a very complete and rigid modeling of the environment, even if completely separated from it, there has been a shift towards an apparently incomplete modeling that allows emergent behavior and learning through interaction with the environment. In this work, we will design a semantic map that will be encoded with the fundamental knowledge, to be able to accomplish it's task. Though through interaction with the environment, it will become increasingly proficient is the task's completion. The task will consist of determining the position of objects in the environment using an object recognition module to sense the world, an action planer, and a hybrid semantic map. The goal of the semantic map is to store and process the sensed information into high-level information, that will be later used by the action planer module. For flexibility pruposes, the knowledge database was designed to integrate information of all types so as to be used by all functional modules. The Problog reasoning engine was designed to enable very complete and mutable models of the environment. Several experiments were made in realistic scenarios, using every day objects. The experiments show clearly that the use of the semantic map makes the search process more efficient, after the first interaction with the environment.

Keywords: Semantic Map, Object search, Probabilistic inference, Object Recognition

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

More and more we find a need to endow servicing robots with reasoning capabilities. According to A. Pronobis [1], the “most fundamental competence for an autonomous mobile agent is to know its position in the world.” But in order to accomplish more complex tasks, the robot needs to be able to sense its environment and represent it so that it can know where to go and which paths to take because *”robots that do not know where objects are have to search for them. Robots that do not know how objects look have to guess whether they have fetched the right one. Robots that do not know the articulation models of drawers and cupboards have to open them very carefully in order to not damage them. Thus, robots should store and maintain knowledge about their environment that enables them to perform their tasks more reliably and efficiently.” [2]* Additionally B. Kuipers [3], considers that commonsense knowledge cannot be achieved without spatial knowledge and that spatial metaphors are constantly used and they “draw on preexisting spatial knowledge to communicate relationships and processes that would be difficult to communicate otherwise.”(cited from G. Lakoff [4])
Generally it is considered that spatial knowledge can take the following forms:

- Metrical maps. (i.e. geometrical representation)
- Topological maps.
- Set of actions to get from one place to the next.

For most of domestic tasks, some kind of reasoning about the environment is required and, although humans do this seamlessly, for domestic robots it is a challenge that has not yet been answered in full. Semantic maps are a powerful tool to address this issue. Semantic maps originated from the semantic networks proposed by, Collins and Quillian, in 1969, as a mean for storing semantic knowledge before Collins and Loftus generalized the concept to cover arbitrary graphical structures [5]. According to R. T. Hartley[6], semantic networks started to be a tool for representing knowledge and later they began their role in building computerized inference systems. Although literature differs significantly on the definition and composition of these networks, three main attributes can be associated:

- They originate in the conceptual analysis of language
- Have equivalent expressiveness to first-order logic
- They can support inference through an interpreter that manipulates internal representations

Moreover Semantic networks involve:

- A reasoning on knowledge based on concepts and relationships among them.
- They can be represented as diagrams
- Computer representation that allows database-like activity and sound inference using algorithms that operate on these representations.

Although it is often referred that semantic networks are diagrams, it is more accurate to say that semantic networks can be represented by diagrams but are really a interconnection of concepts held by a cognitive agent. This representation is very useful for artificial intelligence as well as in other fields like cognitive psychology.

This description of semantic networks and their strong suits shows the advantages of maps like an object oriented semantic map. However it is insufficient for a domestic robot since it does not allow an efficient framework for the use of metric information of the environment like the use required by the navigation task. Hence an extended definition of semantic maps from A. Nüchter's "Towards semantic maps for mobile robots" [7] will be used:

> "A semantic map for a mobile robot is a map that contains, in addition to spatial information about the environment, assignments of mapped features [and functional aspects] to entities of known classes. Further knowledge about these entities, independent of the map contents, is available for reasoning in some knowledge base with an associated reasoning engine."

The majority of maps proposed so far have been for navigation. These enable robots to estimate their location in the environment and to check if a certain destination is reachable and how it can be reached. This kind of map has been optimized for this purpose, but maps for other purposes such as object search may need to store different information or the same information in another way. In short, there are different types of maps:

- Metric Map - A metric map is an accurate, low-level geometrical representation of the environment.
- Topological Map - A topological map is a map obtained through discretization of continuous space into areas called places, these places are linked by paths. This type of map describes the connectivity between places.
- Conceptual Map - A conceptual map is a graph describing the relations(annotated links) between different concepts(nodes) of conceptual knowledge.

Over the years competitions where created to evaluate the state of art and put into perspective the level of performance of robots in a realistic setting outside of the comfort of a laboratory's controlled conditions. A fine example is the RockIn@Home competition, it aims at "[aiding] in the transition from the lab to the market"[8], by providing Testbeds and Task Benchmarks for researchers to evaluate the impact of new features in the actual performance of the robot. Since the Institute for Systems and Robotics at the Superior Technical Institute from the University of Lisbon put together a team called SocRob@home to participate in these challenges and competitions and one of the Task Benchmarks is semantic mapping, there is already in the institution a testbed where the developed system can be properly tested in a realistic environment.

1.2 Problem Statement

A common sign of intelligence is the ability to learn. Thus, a recent question in AI has been the impact that embodiment has in learning. "The world is what we can make of it" (Paul Brady). This means that the robot learns about its environment by interacting with it, thus extending its capabilities autonomously. According to Wooldridge [9]:

> "Brooks also identifies two key ideas that have informed his research.
>
> (1) Situatedness and embodiment. 'Real' intelligence is situated in the world, not in disembodied systems such as theorem provers or expert systems.
>
> (2) Intelligence and emergence. 'Intelligent' behaviour arises as a result of an agent's interaction with its environment. Also, intelligence is 'in the eye of the beholder' - it is not an innate, isolated property".

Ideally the robot should have sufficient tools to be able to infer new meanings and develop courses of actions that would enable him, through experimentation with the environment, to decrease uncertainty on inferred meanings. Let us consider the following example: a robot can have a detailed and optimized pre-programed action plan for openning a certain cupboard door, or it may try to interact with it by pulling in every direction to acertain which is the direction that will open the door. Of course the first will be much more efective, but it will fail when faced with a different cupboard. The second approach may take much more time in completing the same task, but when faced with another type of cupboard, it will still be able to open its door. This type of reasoning, where it is best to model very simple behaviours and let the details of its execution be refined, through interaction with the world, is the principle of embodiement. On a more practical note, how can a robot infer semantic knowledge, from concepts acquired through the teachings (verbal or written instruction) of a supervisor agent, or through interaction with the environment and reasoning upon this knowledge, to be able to validate and improve action plans for complex high-level tasks?

1.3 Objectives

Design and implementation of a system capable of acquiring, representing, and using semantic knowledge to produce a semantic map and, having other modules, use this semantic map to improve the robot's efficiency on completing domestic tasks.

1.4 Assumptions

For this work, a set of aspects need to be considered to mimic, as reliably as possible, a domestic environment. A dynamic environment is to be assumed. When considering a domestic robot, the environment can have people moving around and objects that are moved around. We must also assume that we will have a perfect localization at all times, which implies having an accurate metric map. Regarding the robot, we assume that it will have capabilities for autonomous navigation, identifying objects, and human robot interaction (HRI). It should be noted that, although preferably HRI would be verbal, this is not a requirement, since a simple version of this functionality is sufficient.

1.5 Contributions

The implementation of a semantic map, in a social robot, aims to enhance the capabilities of each and every functional module, by using the high level information acquired by one functional module in another. With this work we aim to show that through the integration of high level information acquired by the object recognition module as an initial belief state of the world for a POMDP based decision making module we can increase the performance in the object search task. A description of the design and impementation will be made with a special focus on the object recognition and semantic map developed.

Chapter 2

State of the art

2.1 Types of Semantic Maps

The domestic robot considered is an autonomous mobile robot and, thus, requires navigation, which cannot be accomplished without a map. The mainstream maps for navigation are metric, topological, and hybrid metric-topological. However, this results in insufficient level of spatial awareness and general semantic knowledge. For this reason, a component of the general map will be dedicated to storing semantic knowledge, either as an extension of the navigation map, or by using a hybrid approach. Below, different types of semantic maps will be described.

2.1.1 Metric

Metric maps are a geometric representation of the environment, that allows for fast generation of collision-free optimal paths and provides accurate localization [10]. The most common geometric map is the occupancy grid. However, it does not scale well with increase in precision and large-scale space, due to memory requirements [11]. This type of map is also hard to create and maintain, because of inaccuracies in robot motion and sensor reading. However, several techniques, using Kalman filter or particle filter0 approaches, can deal with these problems providing for accurate localization. Semantic metrical maps consist of labeled 2D [12] and 3D [11, 13] metric maps. This representation, although useful to show the effectiveness of robot tasks such as object detection, does not provide a good basis for advanced reasoning and symbolic solving [10].

2.1.2 Topological

Topological maps, normally represented by connected graphs, provide a compact description for a collection of places connected by paths [14]. It is a list of significant places, connected via arcs usually annotated with the information to navigate from one place to the next. It describes the connectivity between places[15]. Topological maps have an important strength, consisting on generating all possible topological maps from experience and testing them for consistency. This can provide formal guarantee that the correct map is generated and never discarded [14]. Semantic topological maps consist of adding, to each node, semantic labels characterizing the place. This representation can be used to solve abstract tasks, but it's not ideal for navigation purposes.

2.1.3 Conceptual

A conceptual map is a semantic map in nature. It relates concepts giving them meaning. Although, for a mobile robot, it is not useful on it's own, since it doesn't contain explicit spatial information of the environment making it impossible to use navigation. However, the maps are very useful to enable reasoning, endowing the robot with some "social skills". These skills can be used in HRI. This representation, if extended with probabilities, allows for intelligent decision making. In 2011, M. Hanheide et al. [16] presented a probabilistic conceptual map, "combining general purpose and contingent spatial knowledge in a single structure together with processes for creating, maintaining and reasoning with it."

2.1.4 Hybrid

All of the maps discussed above have advantages and drawbacks. Normally, one representation's advantage is another's drawback. With this in mind, using a hybrid representation is a natural choice. Several kinds of hybrid maps for service robots exist and, naturally, not all of them will be discussed.

In 2004, Kuipers et al. [14] reported an approach using a hybrid metric-topological map that could represent large-scale space, without the specific drawbacks of each map type. It consisted of using a topological map to describe the environment and, when using navigation, making the general path planning on the topological map and using a local perceptual map for obstacle avoidance and small scale navigation.

Later, in 2012, A. Pronobis et al.[17] presented a semantic map approach that in order to deal with a dynamic environment wanted to make spatial knowledge abstract. With this in mind he devised a layered structure comprised of a place layer(i.e.topological map) where he created in its unexplored space hypothesized places called placeholders to be able to reason about unknown space, of a sensory layer where the robot stores the geometrical map of it environment, a categorical layer containing shape models, object models and appearance and a conceptual layer containing common-sense knowledge representation. Additionally a probabilistic conceptual map is used to permit uncertain spatial reasoning, this map is represented as a chain-graph model.

Spatial Semantic Hierarchy

Proposed in 2000, Kuiper's Spatial Semantic Hierarchy (SSH) [3] depicts *"knowledge of large-scale space with four different representations:1) control laws for reliable motion among distinctive state (dstates) xi; 2) causal state-action-state schemas hx,a,x0i and relations view(x,v) between a state and its observable view, abstracting the continuous world to a deterministic finite automaton; 3) a topological model consisting of places, paths, and regions explaining how the distinctive states are linked by turn and travel actions; 4) local metrical information about the magnitudes of actions, the lengths of path* *segments, and the directions of paths at place neighborhoods"*[14]. The apparent heterogeneity of the cognitive map to different observers is proposed by the Spatial Semantic Hierarchy as real feature of the phenomenon and the source of the cognitive map's flexibility, power and robustness [3]. This approach was later extended using local metric maps and global topological maps. In this extended approach metrical mapping methods create and store a local perceptual map of each place neighborhood. At the control level gateways, where control shifts from motion between place neighborhoods to localization within a neighborhood, are identified. The description of the local topology of the place neighborhood can be made from the analysis of the set of gateways in a local perceptual map which simplifies the construction of the topological map.[14]

2.2 How to build the semantic map

2.2.1 Object recognition

Object recognition can be a useful tool for making a semantic map since it can add objects to these maps and maybe even evaluate certain characteristics of the object and insert these into the conceptual map. A similar approach was reported in 2011 by M. Beetz where " a detected object leads to the creation of a *has-object* relation for specific instance the robot was looking for." Later in 2013, identification of objects and doors was used to form a global topology map an object associated map was formed from the relationships of the objects in the room.

2.2.2 Scene recognition/classification

Place categorization based on appearance has a considerable amount of research done on it because "a large share of semantic description of a place is encoded in its visual appearance"[16], however it is not used often[17]. This tool can help to build the semantic map by creating new semantical connections between a place node in a topological map and it's concepts in the conceptual map. This was done in A. Pronobis[1, 17] although this was not the only tool used to build the semantic map in his latest work.

2.2.3 Mapping and Topology Maintenance

Mobile robots use simultaneous localization and mapping(SLAM) algorithms to make and maintain both metric and topological maps[16]. Mapping and the use of SLAM algorithms is a very common and researched topic in robotics. Although the mapping process on it's own does not create new conceptual relations, it is fundamental to acquiring spatial knowledge and through topological maps can endow the robot with some spatial awareness.

2.2.4 Inference

A big advantage of having a conceptual map is to be able to reason on it and to be able to make new relations between concepts, since "relation in the conceptual map are either predefined, acquired, or inferred, and can either be deterministic or probabilistic"[17]. Depending on the representations chosen for the components of the semantic maps approaches may vary. In 2008, A.Nuchter and J. Hertzberg[7] presented a work using Prolog for logical reasoning and inference,this however is a rigid method and does not take into account uncertainty. Later in 2011, M. Hanheide et al.[16] reported a system where the conceptual relations are represented in the form of a chain graph in order to support Bayesian inference. These chain graphs permit modeling of both "directed" causal (i.e. *is-a* relations) and "undirected" symmetric or associative relations(i.e. connectivity) since they provide a natural generalization of directed(Bayesian Networks) and undirected(Markov Random Fields) graphical models [16].

2.2.5 Speech recognition

Just in the way that when you hire a new maid, you need to give her instructions and information on were things are and how you want her to organize the house. This is also considered a valid approach for a service robot, the interpretation of natural language can be a source of conceptual knowledge[18, 19], in which case the robot relates the information given to him in natural language with its internal representation of the environment.

2.3 How to use the semantic map

2.3.1 Speech Recognition

In order for a robot to interact with a human, he must have a way of transforming his perceived data into human compatible concepts. Some researchers say that robots also need *social skills* to be able to interact with humans[20]. This means that systems that are involved in human robot interaction through written or verbal speech need to be able to respond to high level commands thus requiring semantic knowledge[21].

2.3.2 Object recognition,manipulation and search

In a dynamic environment like the domestic setting we propose to do, the robot will have to adapt to searching for an object that has been moved. Normally the reason for moving an object is related to the object's nature or can be modeled accurately with probabilities. This is an approach that researchers are starting to consider[22, 21]. It is also important for a robot to be able to interact with its environment. In that sense, researchers have been looking into the possibility of robots recognizing an object's *affordance* [23, 24]. In 1979, psychologist J. Gibson defines affordances as the actionable properties between an agent and the environment, therefore depending on its motor abilities. One approach to infer the affordance of an object with vision and preexisting knowledge as was done by Afonso Gonçalves in [25], where the best tool for a given goal could be selected according to its affordance.

2.3.3 Navigation

I order to execute a navigation task, the robot must know where he is, where he wants to go and how he can get to his destination. For humans these concepts are normally described as semantic labels[26]. When a human gives a robot a navigation task, the robot must be able to relate semantic concepts to locations in its internal representation of the environment. In this case, the navigation module must use the semantic map in order to understand its goal location or, in a more ambitious case, to understand the instructions given by the user. When giving instructions for navigation, humans tend to give very imprecise metric related instructions and thus normally use topologically related instructions. This will naturally have semantic references to places and will require a human-compatible system to associate these concepts with metric location,through direct relations or through reasoning, which are preferred for robot navigation. A semantic map is also very useful for navigation in dynamic environments, like the domestic one, where for example furniture may be moved and different factor may lead to unexpected optimal paths. If when mapping the robot is able to recognize which parts of it's environment are subject to change and which are static, then the robot can increase it's localization robustness and have more information for making navigation plan, for example it can choose to go along a longer path because it is usually less crowded hence being faster. [20] clearly states that "[s]emantic path planning is proceeded to describe its outstanding advantage on extending planning capabilities by reasoning about semantic information and improving planning more efficiently in a larger domain." In a very similar approach [16] presents a Switching Continual planner that uses a starting belief-state description compiled from a probabilistic conceptual map in order to perform the navigation plan.

Chapter 3

Theoretical Background

3.1 Symbolic logic

Symbolic logic is the study of symbolic abstractions that capture the formal features of logical inference.[27] Symbolic logic is often divided into two branches: propositional logic and predicate logic. In propositional logic each possible atomic fact requires a separate unique propositional symbol. If n people and m locations then n moved from m1 to m2 requires n*2^m different symbols. Predicate logic includes a richer ontology, it include objects(terms), properties(unary predicates on terms), relations(n-ary predicates on terms) and functions(mappings from terms to other terms). It is more flexible and has more compact representation of knowledge.[28]

3.1.1 First Order Logic

First Order Logic, also known as first order predicate calculus, is a formal system used in several fields such philosophy, linguistics, mathematics and computer science. The difference between first order logic and higher order logic is in this case predicates cannot have other predicates or functions as arguments. In traditional grammar, a predicate is one of the two main parts of a sentence the other being the subject, which the predicate modifies, similarly in logic the predicate is a function that for a given subject outputs a boolean value. In general terms the predicate modifies or defines the properties of the subject.

3.2 Logic Programming

Logic Programming is a type of programming consisting of facts and relationships from which the programming language can draw a conclusion. Instead of providing the procedure for modifying the program states and variables like in imperative programming languages, in logic programming, the computer will on its own derive answers from the userprovided facts and relationships through logic inference. One of the most widely used logic programming languages is Prolog.

Prolog is a full-featured programming language, where running a program implicates proving a theorem, since the programs consist of logical formulas, so in order to run a prolog program, one must pose a query. A program consists of a set of procedures, a procedure in turn consists of clauses. Each clause is a fact or a rule making a problog program a sort of relational database. In Prolog's syntax, there are variables, objects, predicates and functions organized together as facts and rules. Rules however are a special case for they include the operator $:-$ which should be read as an implication.

- Object $->Mug.$
- Predicate$->breakable(Mug).$
- Function $->append(kitchenobjects, Mug, newkitchenobjects)$
- Rule $->breakable(X):-Glass(X).$
- Fact$->Glass(Mug).$

Prolog is without a doubt a very powerfull theorem prover however in real life there is lot of uncertainty and it is something that we deal with every day. With the push for autonomus robots it has been increasingly apparent that ways of dealing with these uncertianties are required.

3.2.1 Probabilistic Logic Programming

Over the past twenty years an increasing number of probabilistic logics has been developed, although there are still only a few real-life applications of these logics, maybe because of a multitude of factors such as the restrictions they impose, their assumptions may be too strong, their solvers may be too limited or too slow. They include PHA, PRISM, SLPs, MLNs and probabilistic Datalog.

Problog

Problog is a probabilistic extension of Prolog taking advantage of all of it's power but keeping it as simple as possible. It consists of probability labelling all clauses and making them mutually independent. Complex real-world applications have already been given where, Problog is used in mining large biological networks whose edges are labelled with probabilities. The key in this application is the independence of probabilistic links. The algorithms used to compute the success probabilities of a query were implemented using prolog's inference with Binary Decision Diagrams.

Regrading the syntax, problog aims and maintaining much of the structure of prolog. The anotation of the clauses is made through the :: symbol at the beggining of the fact or rule. It must be remarked that if omitted it is considered to be 1, and thus it will have the same meaning as in Prolog. Another very interesting feature of problog is the evidence operator, with it, it is possible for one to specificy that a probabilistic fact defined in the program has happened. It works much in the same way as a fact would in prolog.

Chapter 4

Semantic Map for object search in domestic robots

4.1 Description of the components of the problem

As explained above, the semantic map should be constructed using as many data sources as possible and used to enhance all the functional modules of the robot. Due to the complexity of this tool, only functionalities specific to the object search where considered although the design was made as flexible as possible to be able to accommodate more functional models. For a robot to complete the task of fetching an object(a multitude of steps are required), some information regarding the environment is required otherwise, the robot will have to engage in a brute force search. In order for the robot to be efficient in the search process, we give it some a priori general knowledge and we the teach him what that abstract knowledge represents. The grounding of the abstract knowledge is what makes the robot able to reason upon the abstract information given and act based on it's conclusions(or findings), this approach of interconnecting the real world information and the abstract knowledge is the basis of the embodiement theory that so many researchers in the field consider as being the cornerstone to advance artificial intelligence [29]. In order to use the robot's sensory information, the robot must first acquire it and store it conveniently. In this approach a flexible representation of the world was made to enable several types of reasoning. For the reasoning engine, an integration with a platform, *Problog 2*, was designed and implemented to accomplish the specific task at hand.

4.2 Structure of the Semantic Map

The semantic map can be divided into two main parts, the knowledge base and the reasoning engine, as can be seen in figure 4.3.

The first part, the knowledge base is where the information acquired by the functional modules is stored, be it an abstract knowledge concept or the information regarding a specific instance in the environment.

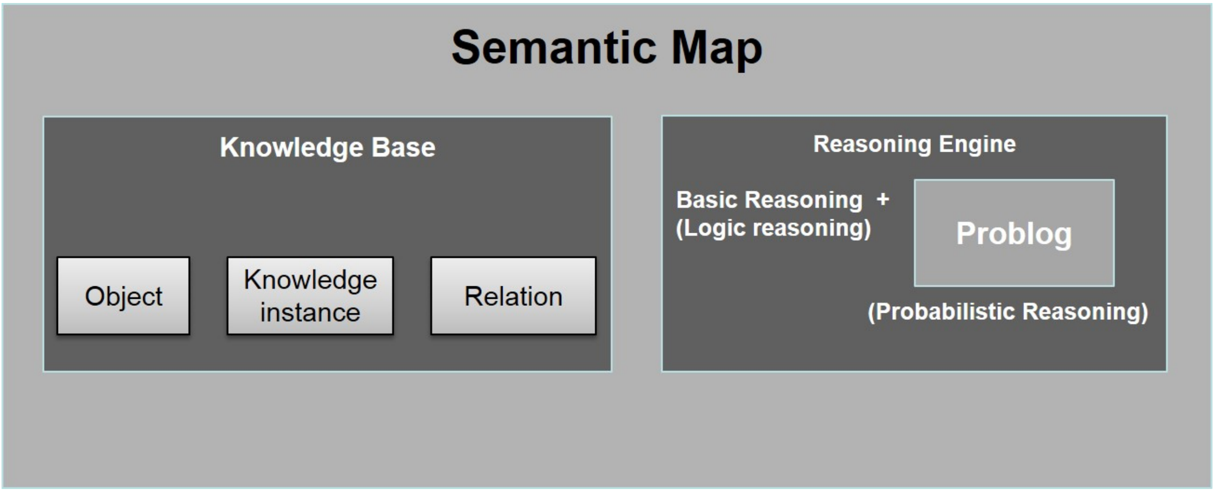

Figure 4.1: Semantic Map Structure

The second part, the reasoning engine is where the low level information is converted to high level information and where the high level information is processed to create the initial belief state that will later be used by the other modules.

4.2.1 Knowledge Database

The knowledge database can accommodate three types of knowledge:

- A knowledge instance is used to represent common sense information regarding a specific concept; (Examples: - sodacan - table)
- An object information type will be the physical representation of a concept; (Examples: coke - Volume(6.6) - BaseArea(0.66) ; cereal - Volume(16.6) - BaseArea(1))
- Relations represent the interconnections between knowledge instances or actions the robot can take; (Examples: SodaCan-IsA(MovableObjet) ; MovableObject-IsA(Object))

This may seem like an obvious and overly simplistic representation but it is very powerful for it can give the robot a basis to understand indirect referencing and can extend the existing functionalities by introducing the notion of categorization. With recourse to our everyday day life we can see just how fundamental this notion is. Consider the following scenario, upon arriving into a room where we have a table with an cup on top, someone points in that general direction and identifies something. Without any previous information regarding what we are looking for we will not be able to understand what was referred. However if we know that the reference was to a piece of furniture, we can infer that the person was referring to the table. In this simple reasoning process categorization was used to remove uncertainty on what we were being told. There are two types of relations considered:

- Unidirectional relations where one can specify for example hierarchy between knowledge instances;
- Bidirectional relations for specifying actions the robot can take in a cause effect manner. This feature was implemented to allow for a rudimentary action planning and interaction pattern. For

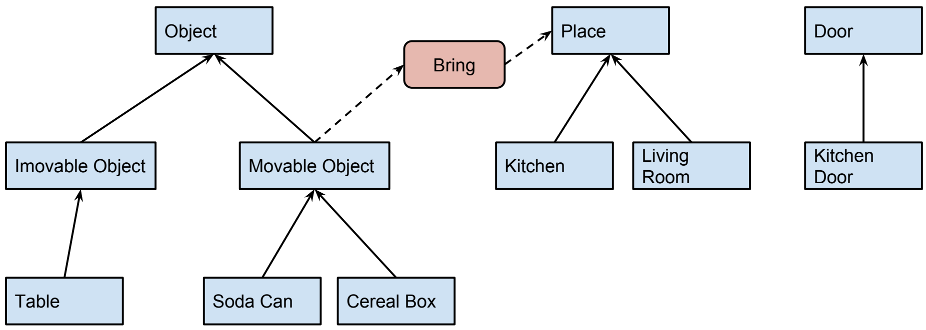

Figure 4.2: Knowledge Base

simplicity only actions with one cause are defined although more complex actions can be achieved by chaining simpler actions.

4.2.2 Reasoning Engine

The reasoning engine in the semantic map is a fundamental feature for it is the basis for the intelligence of the robot. Simple verification of action planning can be accomplished through the analysis of bidirectional relations and making use of the hierarchical dependencies between concepts as can be seen in figure 4.2.

However for the task at hand this structure alone was insufficient because it could not deal with several key aspects:

- Uncertainty in the detection modules,
- Different priority depending on when an observation was made,
- Limitations inherent to the sensoring equipment.

For this reason the probabilistic reasoning platform used was Problog whose syntax and semantics can be seen in [30]. We used several probabilistic graph models to model the environment:

- Bayesian Networks - These models enabled us to describe the whole environment with some constraints like: - No object can be in two places at once, - The reason an object has not been seen, can be because another was in front of it. However it was not possible to integrate the observations collected since we were not accounting for time in this model and the inclusion of evidences of an object that was seen in two places generated inconsistencies.
- Hidden Markov Models - With these models it was possible to account for the time between observations but describing the whole environment was shown to be computationally very expensive. It was impossible to consider more then 3 time steps.

Taking advantage of the high level of integration between the Database and the Problog reasoning engine, we managed to overcome the weak points of the previous models, making it possible to consider the whole environment and take into account the time between observations. It was achieved by making

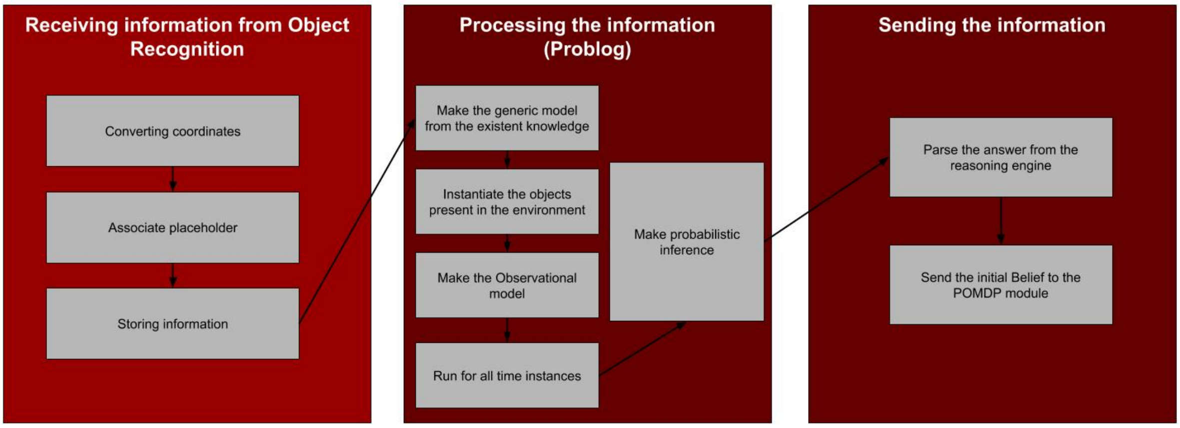

Figure 4.3: Semantic Map Workflow

a Bayesian Network model of the world per time-step and to use the elapsed time between time-steps to weigh in on the influence of observations in the current belief of the world. It was also possible to model other interesting features such as occlusion between objects and having the mobility of an object influence the obervation model of the robot, as can be seen in Appenddix A.2.

4.3 Communication

The communication interface of the semantic map is a fundamental feature and needs to be as flexible as possible to accommodate the different kinds of information acquired by the functional models but also designed to be as clear and of as direct interpretation as possible. The communication method designed for receiving information from the functional models is a topic, that will be referred hence forth as the teaching topic. The teaching topic is fundamental in the construction of the semantic map. It is meant to be used in the all of the functional modules and so the message type has all of the types of knowledge available in the knowledge base. The structure of the message is detailed in B.1.
In order to facilitate the current application, a service was implemented to query the belief state of the environment, see Appendix B.2. This initial belief state will have the result of the processing of all of the grounded information in the knowledge base by the reasoning engine.

4.4 Integration with the functional modules

Making the integration of all modules is of fundamental importance to test the functionalities developed and complete the object search task.The procedure for comunication is shown in Figure 4.4. When starting this task the semantic map module comunicates the initial belief to the decision module when starting each run, the decision module sends the recognition request and receives its outcome. Additionally all objects successfully found are comunicated to the semantic map.

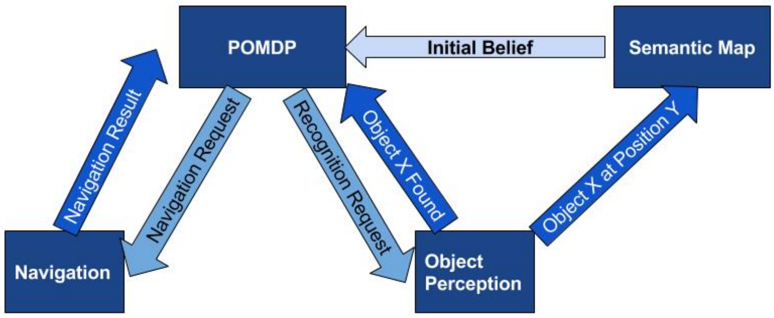

Figure 4.4: Task Execution and Information Flow

4.4.1 Object Recognition

In order for the semantic map to have updated information on the environment, every time the object recognition module finds an object the result is sent to the semantic map module which in turn updates it's knowledge base. This information is sent in *base_link* coordinates through the teaching topic and will be treated by the semantic map as an observation.

Dealing with observations

To model an observation we use a probabilistic fact whose weight depends on the time difference between the last observation of the object and the current one and the mobility factor associated with each object:

- $P_(x, y)(T2|T1) = 0.6 + 0.1 * e^{-(T2-T1)*M/10}$

- $P_(x, \overline{y})(T2|T1) = 0.3 - 0.1 * e^{-(T2-T1)*M/10}$

Placeholder alocation

Whenever the object recognition module identifies an object, the only information supplied to the semantic map is the class of the object and the position of the object in relation to the base_link reference frame. The semantic map will convert these coordinates into the map reference frame, as can be seen in figure 4.5, and acertain on which placeholder the object was detected. This step is instrumental in making the bridge between the object recognition module and the POMDP module and makes use of environment specific information stored in the semantic map,the placeholders locaton in map coordinates.

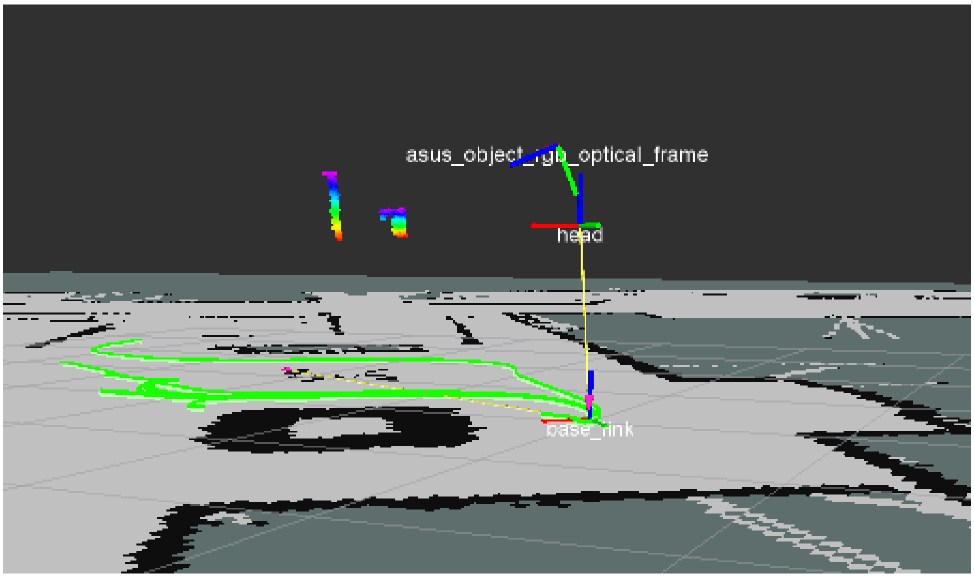

Figure 4.5: Point Cloud of objects recognized on the map

4.4.2 Decision making module (POMDP)

Using the information gathered about the environment and processing it with the reasoning engine, the robot is able to formulate a belief on the state of the world at a given instant. To test the validity of this belief, this information is used as the initial belief state for a POMDP based decision module. This module was developed by Tiago Veiga a Post-doctural student from the ISR department specifically for accompishing the object search task using semantic map information. Since the strong suit of this approach is to learn by experiencing, the semantic map will feed the information to the decision making module but continue to collect information as the robot performs the task. This way the next time it is called, it will give an initial belief based on the base information and the newly acquired information.

Chapter 5

Object Recognition

Object recognition is one of the most fundamental features for a domestic robot. Several approaches were considered and implemented, including RGB object detection but ultimatly 3D object detection yielded the best results. Our Object recognition module is base in the 3D recognition framework of the PCL library [31].

5.0.3 Description of previous implementation

The first version of the object recognition was based on a RGB object detection. Our module was comprised of two stages, the first was the initialization stage where all of the models for the objects were loaded and processed, the second stage was the acquisition of the images from the RGB camera and the recognition.

In the initialization stage the module loaded a series of pictures corresponding to each object, for each of them it calculated key points and the corresponding descriptors. A database was thus created with all of the key points and descriptors for each view of the object.This step was a lengthy one but was done only once when the module was initialized.

In the second stage of the process, upon a request from the state machine a picture of the scene would be acquired and processed. The image of the scene would undergo the same process as the views of the objects, the key points were calculated and their corresponding descriptors as well. The key points from the scene were matched to the key points of every view of each object and the view with the highest number of matches is selected. To avoid false positives the homography matrix between the two sets of coordinates is calculated and the an error value calculated. The validation of the image was done by comparing color histograms of the object in the scene to the view of the object that had the best match. This method although proven to work in theory was ineffective in practice, the reasons for this lack of success were the low resolution of the camera, the dependence on lighting conditions, the high amount of views required for each object, the low amount of keypoints for single color objects. This was however an effective recongition method for flat objects with a fixed pattern like a picture frame. In order to chose the most appropriate descriptors, several tests were made in real conditions with different descriptors and SIFT proved to be the most accurate.

5.0.4 Description of current module(3D Pipeline)

The 3D object recognition Pipeline is comprised of two modules, a training module and a recognition module as can be seen in figure 5.1.

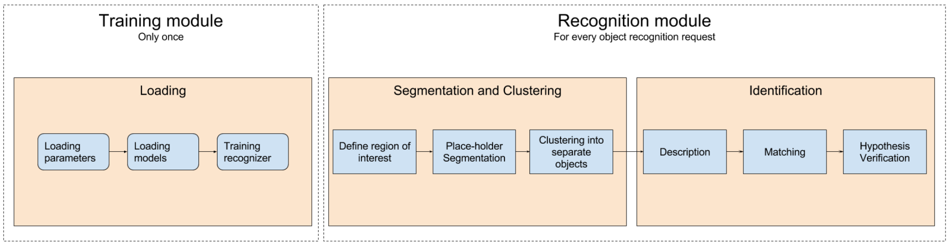

Figure 5.1: 3D Object Recognition Pipeline

Training module

The training module imports models for an object class in binary PLY format. These models are then rotated and converted to point clouds from different views for each view several keypoints identified and corresponding our-cvhf descriptors are extracted.

Recognition module

The recognition process is comprised of three main steps:

- Loading of the information required by the module,
- Making the scene segmentation into object clusters
- Identifying clusters of objects.

In the Loading stage, the module will load all the models available to the recognizer as well as specific information needed for the segmentation and coordinate conversions. This involves receiving several user defined parameters and some coordinate transform information.

After this step comes the segmentation of a scene's point cloud. In this step the module will have to use either the tabletop segmentation for when objects are in a flat surface in front of the robot or the 3D background subtraction used in case the objects in different shelves of a book case for example. In either case the module will filter the area of interest of the scene and apply a clustering algorithm to the remaining point cloud. Following this process, we can extract the position of the cluster and thus the object.

However the module will have a series of clusters that need to be classified, for this part a recognizer trained with the models previously processed will present a set of most likely correspondences

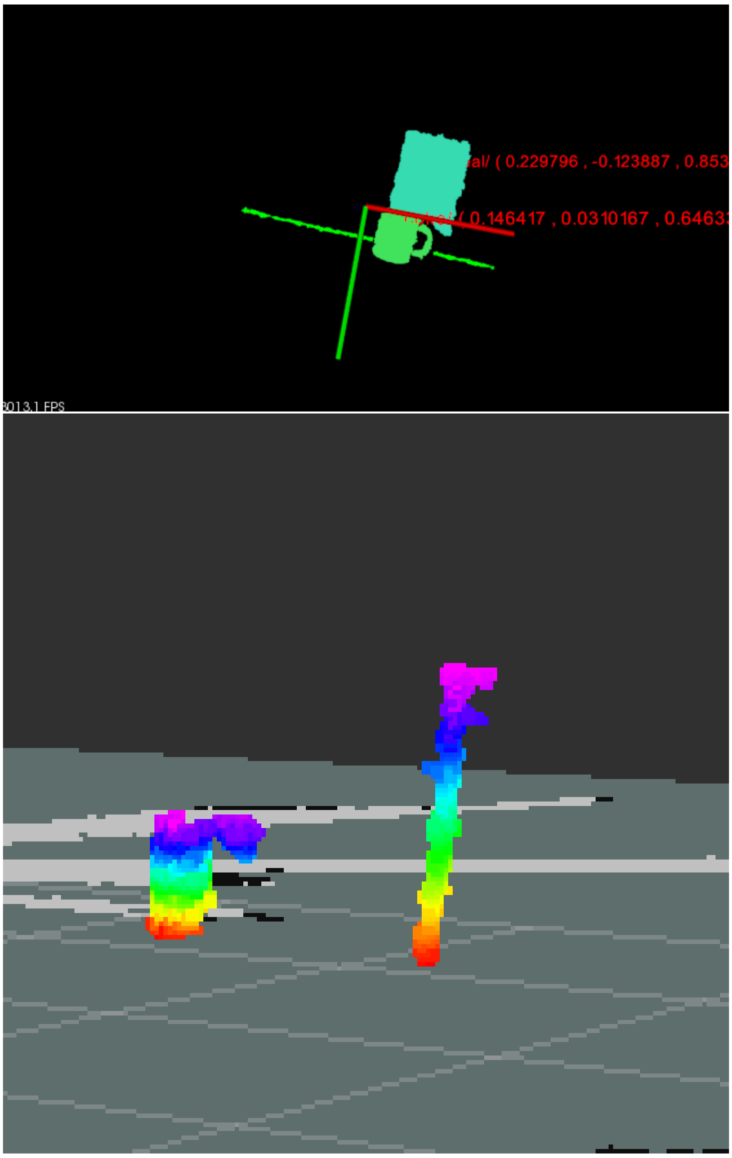

Figure 5.2: Objects recognized

discriminating classes and models within that class and an error value (mean value of the distance in the descriptor space) associated with that match, the class and model of the object is considered to be the one with the lowest error value.

5.0.5 Description of the upgrades made to the 3D object recognition pipeline

To enhance the accuracy of the segmentation process, a generalized placeholder subtraction was developed since the typical tabletop detection does not yield satisfactory results when the robot is trying to detect objects on a placeholder like a bookshelf. The general placeholder is very similar to an image's background subtraction. This segmentation method, described in Figure 5.3, requires a previous point cloud model of the placeholder for the object. When the robot is facing the object's placeholder, from a predefined perspective, the model of the place holder is aligned in the scene and then subtracted to it. The method for aligning the model with the placeholder itself was initially a regular ICP but due to the high computational cost of this approach for a large model and to improve accuracy, an ICP with RANSAC that used only the 3D descriptors of the scene and the model was used. This approach successfully allowed for the detection of objects in a bookcase.

Figure 5.3.1 Scene

Figure 5.3.2 Matching

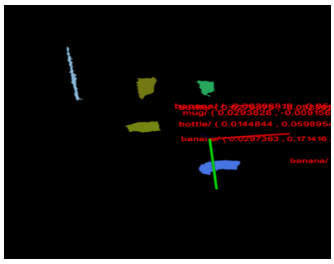

Figure 5.3.3 Clustering

Figure 5.3: General segmentation method

Chapter 6

Results

The experiments were made with the robot described in [32], with the addition of a Asus Xtion Pro Live RGB-D camera.
In order to test the functionality of the system developed 3 experiments were conducted each of which comprised of several runs. In each run the robot will attempt to find two objects placed on the table and counter in the kitchen seen in figure 6.2. At the beginning of the run, the robot, upon receiving the start command will process the information stored in the semantic map from previous runs and will create a initial belief that will be passed to the POMDP decision making module who will make an action plan and proceed to move about in the environment, an example of a robot path is the green line in the figure 6.1.

6.1 Test conditions

In order to benchmark the functionalities of each of the modules and the level of interaction achived between them, some restrictions were put in place.

6.1.1 Semantic map

In order to make independent experiences, the knowledge acquired in an experience will be used in subsequent runs but will not be used in for other experiences. The knowledge used by the semantic map to calculate the first initial belief supplied to the decision module is the same in all experiences and is described in Appendix A.1.

6.1.2 POMDP based decision making module

For simplicity in analysing the decision module to acertain the validity and usefullness of the information provided by the semantic map, a restricted model of the environment was made. The understanding of these constraints is key in analysing the behaviour of the robot. In this model, there are only two models and two place holders, given the difference in volume between the object only the cereal box can occlude

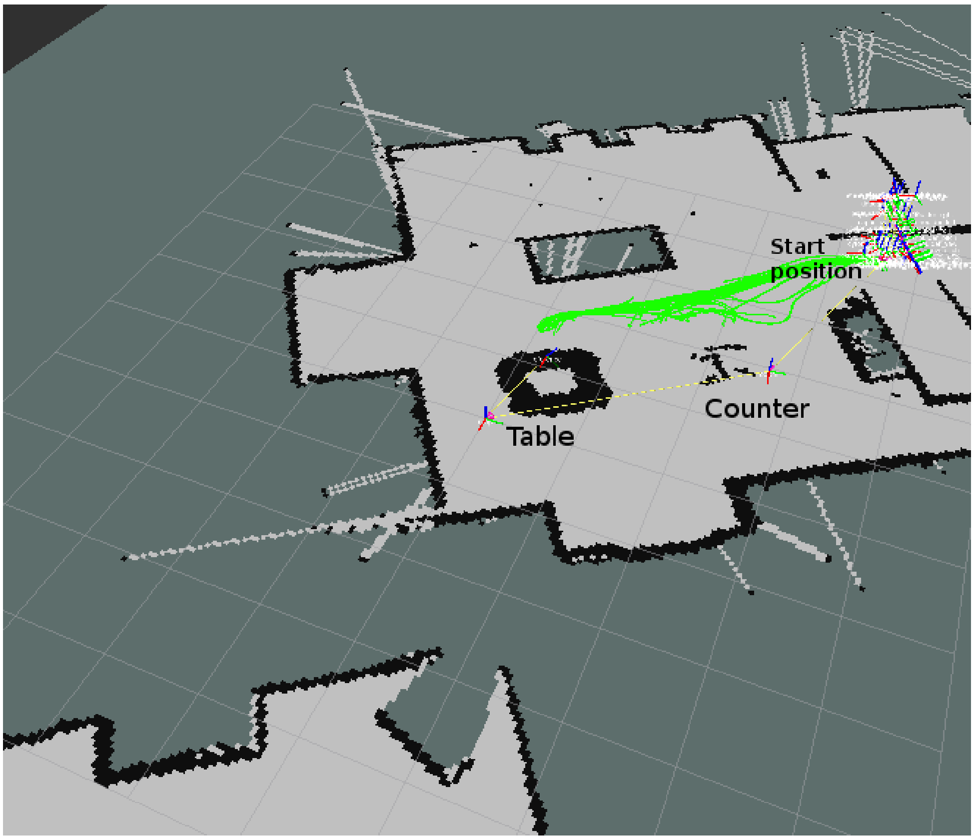

Figure 6.1: Top view of the scenarion and the robot paths planned.

Figure 6.2: Objects recognized

the coke and both objects are in the environment. The last restriction can seem overly simplistic but in fact, since we can model the objects being out of the environment as being in a placeholders where no observations can be made.

6.2 Experiments

6.2.1 Experiment 1

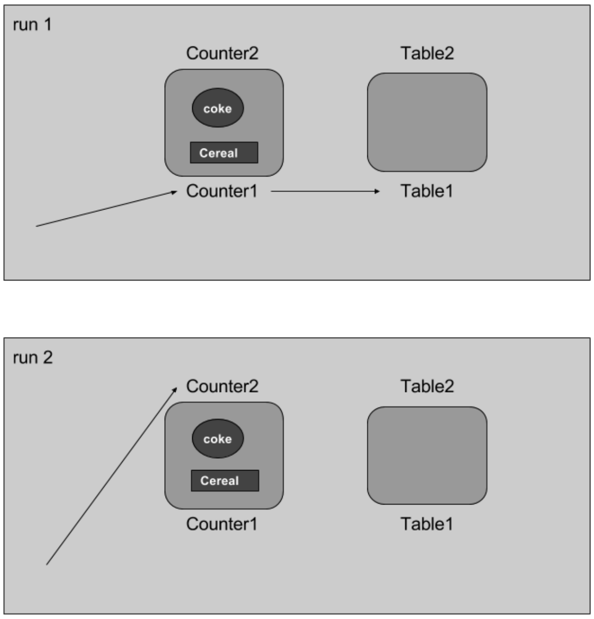

Figure 6.3: Object placement and robot paths for Experiment 1

6.2.2 Run 1

The probability distribution of the objects' placement is uniform, thus the POMDP module will chose the observation point that is closest. As can be seen by the object disposition in 6.3.

6.2.3 Run 2

Since the object Coke was no detected in the previous run, the semantic map has the same probability distribution along the placeholders for that object. The cereal box however, since it was seen on the counter, has a higher probability of being on the counter.

6.2.4 Experiment 2

Run 1

As is normal for all first runs, the probability distribution is uniform for all objects over all placeholders. Since only the cereal box was detected on top of the counter, the

Objects \Locations	Counter	Table	Bench
Coke	0.33	0.33	0.33
Cereal	0.33	0.33	0.33

Table 6.1: Initial Belief- Experiment 1 Run 1

Starting POMDP execution				
Loading policy...				
New action:	goCounter1			
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal	oYes
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oYes
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oYes
New action:	goTable1			
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oNo
New action:	doNothing			
Final response:	Cereal:	inCounter,	Coke:	inCounter
POMDP execution completed				

Table 6.2: POMDP execution for Experiment 1 Run 1

Objects \Locations	Counter	Table	Bench
Coke	0.33	0.33	0.33
Cereal	0.50	0.25	0.25

Table 6.3: Initial Belief- Experiment 1 Run 2

Starting POMDP execution				
Loading policy...				
New action:	goCounter2			
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal	oYes
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oYes
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oYes
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oYes
New action:	doNothing			
Final response:	Cereal:	inCounter,	Coke:	inCounter
POMDP execution completed				

Table 6.4: POMDP execution for Experiment 1 Run 2

Run 2

- The initial belief of this run (table 6.7) reflects the sightings of the objects in the previous run, each object was seen three times on their placeholder therefore the distribution of probabilities is analogous.

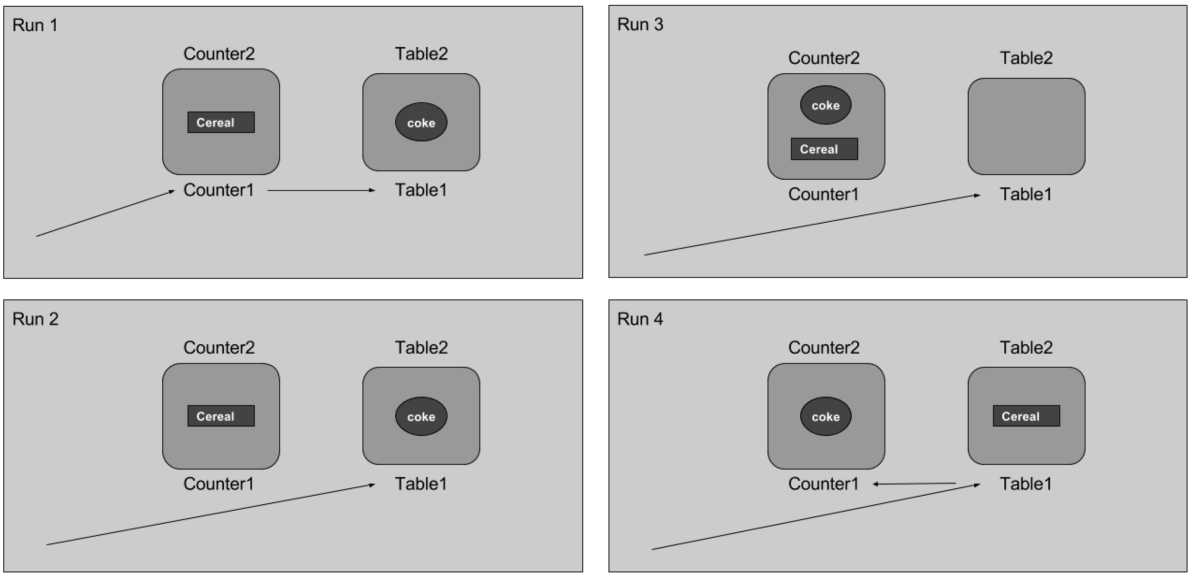

Figure 6.4: Object placement and robot paths for Experiment 2

The decision making module, taking into account the information provided by the semantic map will elaborate the most efficient planning scenario. Given the module's restrictions, the module decides to look first at the table from position 1 and having found only the coke there concludes that the cereal box can only be in the counter. This situation contrasts heavily with the first run of this experiment where the robot had to make use of two observation points because it had no conclusive prior knowledge regarding the object's position.

Run 3

Having sighted only the coke in the previous run, the cereal box has the same probability distribution as before, the other object however, increased the probability of being on the table (table 6.9).
Since the type of probability distribution is the same as before, the behaviour is, as expected, also the same. It proves to be very efficient, since the coke cannot be seen in the first observation stop(table 6.9), given the modelled restrictions, it can be concluded that both the cereal box and the coke are in the counter.

Run 4

- Since in the previous run no objects were observed, the initial belief is the same as in run 3 (table 6.11), however given that the object placement, the robot requires two observation points to conclude on the objects position. The first observation point is the table and only the cereal box is observed and this object may be occluding the coke, the robot goes to the counter since it cannot conclude on teh coke's wereabouts. Having sighted the coke at the second observation point the robot concludes that the coke is on the counter and the cereal box is on the table.

Objects \Locations	Counter	Table	Bench
Coke	0.33	0.33	0.33
Cereal	0.33	0.33	0.33

Table 6.5: Initial Belief- Experiment 2 Run 1

Starting POMDP execution				
Loading policy...				
New action:	goCounter1			
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal	oYes
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oYes
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oYes
New action:	goTable1			
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	doNothing			
Final response:	Cereal:	inCounter,	Coke:	inTable
POMDP execution completed				

Table 6.6: POMDP execution for Experiment 2 Run 1

6.2.5 Experiment 3

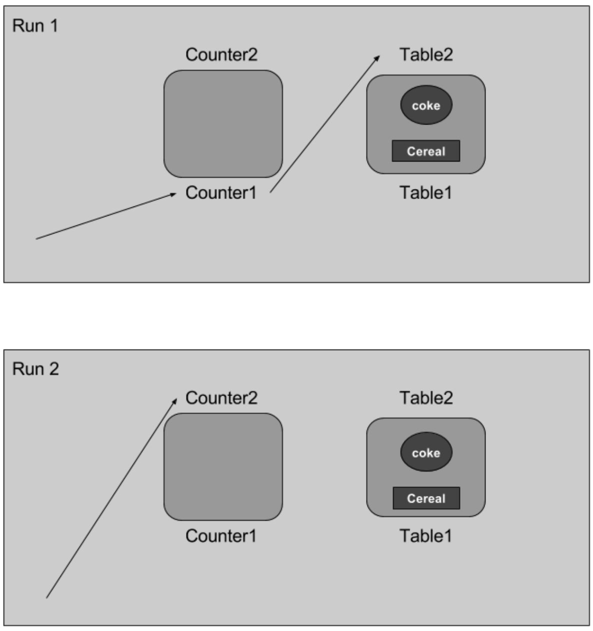

Figure 6.5: Object placement and robot paths for Experiment 3

Objects \Locations	Counter	Table	Bench
Coke	0.065	0.87	0.065
Cereal	0.87	0.065	0.065

Table 6.7: Initial Belief- Experiment 2 Run 2

Starting POMDP execution				
Loading policy...				
New action:	goTable1			
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	doNothing			
Final response:	Cereal:	inCounter,	Coke:	inTable
POMDP execution completed				

Table 6.8: POMDP execution for Experiment 2 Run 2

Objects \Locations	Counter	Table	Bench
Coke	0.03	0.94	0.03
Cereal	0.87	0.065	0.065

Table 6.9: Initial Belief- Experiment 2 Run 3

Starting POMDP execution				
Loading policy...				
New action:	goTable1			
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oNo
New action:	doNothing			
Final response:	Cereal:	inCounter,	Coke:	inCounter
POMDP execution completed				

Table 6.10: POMDP execution for Experiment 2 Run 3

Run 1

Since the initial belief is uniform, the POMDP module decides to go to the closest location, the counter. Since no object is found on the counter, the next observation point is the table where both objects are detected. Even though the cereal box which is behind the coke is found in only 4 of the 6 object recognition attempts, the confidence achieved is enough to conclude on the objects' position.

Run 2

As expected in the probability distribution for the object placement is very high on the table for both objects as can be seen in table 6.15. The POMDP decision module decides to look for the objects on the counter, since they were not found there after only two observations, it concludes that both objects

Objects \Locations	Counter	Table	Bench
Coke	0.03	0.94	0.03
Cereal	0.87	0.065	0.065

Table 6.11: Initial Belief- Experiment 2 Run 4

Starting POMDP execution				
Loading policy...				
New action:	goTable1			
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal	oYes
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oYes
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oYes
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal	oYes
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oYes
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal:	oYes
New action:	goCounter1			
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	doNothing			
Final response:	Cereal:	inCounter,	Coke:	inTable
POMDP execution completed				

Table 6.12: POMDP execution for Experiment 2 Run 4

are still on the table.

Objects \Locations	Counter	Table	Bench
Coke	0.33	0.33	0.33
Cereal	0.33	0.33	0.33

Table 6.13: Initial Belief- Experiment 3 Run 1

Starting POMDP execution				
Loading policy...				
New action:	goCounter1			
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal	oNo
New action:	goTable2			
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oYes
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal	oYes
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oNo
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oYes
New action:	searchObject			
Observations:	Coke:	oYes,	Cereal:	oYes
New action:	doNothing			
Final response:	Cereal:	inTable,	Coke:	inTable
POMDP execution completed				

Table 6.14: POMDP execution for Experiment 3 Run 1

Objects \Locations	Counter	Table	Bench
Coke	0.03	0.94	0.03
Cereal	0.07	0.86	0.07

Table 6.15: Initial Belief- Experiment 3 Run 2

Starting POMDP execution				
Loading policy...				
New action:	goCounter2			
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal	oNo
New action:	searchObject			
Observations:	Coke:	oNo,	Cereal	oNo
New action:	doNothing			
Final response:	Cereal:	inTable,	Coke:	inTable
POMDP execution completed				

Table 6.16: POMDP execution for Experiment 3 Run 2

Chapter 7

Conclusions

In the present work we aimed at developing the different functional modules needed to accomplish the object search task and to prove that through the integration of these modules it was possible to have an efficiency gain. The experiments acomplished show that all 4 modules are fully functional, comunicate seamlessly and show that the processing of low level information into high level information done in the semantic map can in fact extend the capabilities of the decision making module. Since the POMDP based decision module calculates the best action based on the available information, in every experiment, if the object disposition is the same as in the previous run, the object search is faster the following run.This is evident because there is an actual reduction in the number of observation points and recognition requests at each point. Hence it is possible to conclude that the information generated by the semantic map is in fact usefull. It can also be seen that the architecture developed although much broader in scope because it is meant to accomodate the needs of other functional modules, provides the necessary base for the task at hand. With this semantic map module the robot can receive information from any given module as long as it is converted to the format of the teaching topic. It is possible for the semantic map to receive any kind of information while online for new objects, new occurences of objects, new knowledge instances or new actions.

7.1 Future Work

The development of this framework proved to enhance the capabilities of the decision making module taking advantage of having a centralized information storage and processing module. Naturally the next step would be to have more modules contribute to the semantic maps construction and have more modules use it. It's forseen usages include integrating the semantic map into the speech recognition by constructing the grammar from the instances know to the robot and to verify that the commands or teachings do not prove to be inconsistent(example of inconsistencies: the cereal box is in the coke; Please pour milk into the cereal box), further integration with the object recognition to choose autonomously which type of segmentation to use and which model to use given the robot's location and orientation. Concerning the remaining modules developped, further work can be done also in the object recognition module like incorporating new object models online, and making the recognizer use the texture of the object as well as its shape for the recognition.

Bibliography

[1] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *The International Journal of Robotics Research*, 2009.

[2] D. Pangercic, B. Pitzer, M. Tenorth, and M. Beetz. Semantic object maps for robotic housework-representation, acquisition and use. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4644–4651. IEEE, 2012.

[3] B. Kuipers. The spatial semantic hierarchy. *Artificial intelligence*, 119(1):191–233, 2000.

[4] G. Lakoff and M. Johnson. *Metaphors We Live By*. The University of Chicago Press, 1980.

[5] T. Griffiths and M. Steyvers. A probabilistic approach to semantic representation. In *Proceedings of the 24th annual conference of the cognitive science society*, pages 381–386. Citeseer, 2002.

[6] R. T. Hartley and J. A. Barnden. Semantic networks: visualizations of knowledge. *Trends in Cognitive Sciences*, 1(5):169–175, 1997.

[7] A. Nüchter and J. Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.

[8] Description of the rockin at home challenge, 2014. URL `http://rockinrobotchallenge.eu/home.php`.

[9] M. Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[10] P. Buschka and A. Saffiotti. Some notes on the use of hybrid maps for mobile robots. In *Proc. of the 8th Int. Conf. on Intelligent Autonomous Systems*, pages 547–556, 2004.

[11] B. Siemikatkowska, J. Szklarski, and M. Gnatowski. Mobile robot navigation with the use of semantic map constructed from 3d laser range scans. *Control and Cybernetics*, 40:437–453, 2011.

[12] I. Jebari, S. Bazeille, E. Battesti, H. Tekaya, M. Klein, A. Tapus, D. Filliat, C. Meyer, S.-H. Ieng, R. Benosman, et al. Multi-sensor semantic mapping and exploration of indoor environments. In *Technologies for Practical Robot Applications (TePRA), 2011 IEEE Conference on*, pages 151–156. IEEE, 2011.

[13] A. Nüchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, and H. Surmann. 3d mapping with semantic knowledge. In *RoboCup 2005: Robot Soccer World Cup IX*, pages 335–346. Springer, 2006.

[14] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4845–4851. IEEE, 2004.

[15] S. Thrun et al. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, pages 1–35, 2002.

[16] M. Hanheide, C. Gretton, R. Dearden, N. Hawes, J. Wyatt, A. Pronobis, A. Aydemir, M. Göbelbecker, and H. Zender. Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2442, 2011.

[17] A. Pronobis and P. Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3515–3522. IEEE, 2012.

[18] M. R. Walter, S. Hemachandra, B. Homberg, S. Tellex, and S. Teller. Learning semantic maps from natural language descriptions. Robotics: Science and Systems, 2013.

[19] K. Li and M.-H. Meng. Indoor scene recognition via probabilistic semantic map. In *Automation and Logistics (ICAL), 2012 IEEE International Conference on*, pages 352–357. IEEE, 2012.

[20] H. Wu, G.-h. Tian, Y. Li, F.-y. Zhou, and P. Duan. Spatial semantic hybrid map building and application of mobile service robot. *Robotics and Autonomous Systems*, 62(6):923–941, 2014.

[21] A. L. Chiesa. Dynamics aware 3d occupancy grid map with semantic information. In *Advanced Robotics (ICAR), 2013 16th International Conference on*, pages 1–6. IEEE, 2013.

[22] C. Cadena and J. Košecká. Recursive inference for prediction of objects in urban environments. In *International Symposium on Robotics Research*, Singapore, December 2013.

[23] A. Stoytchev. Behavior-grounded representation of tool affordances. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3060–3065. IEEE, 2005.

[24] D. Kraft, R. Detry, N. Pugeault, E. Başeski, J. Piater, and N. Krüger. Learning objects and grasp affordances through autonomous exploration. In *Computer Vision Systems*, pages 235–244. Springer, 2009.

[25] A. Gonçalves, J. Abrantes, G. Saponaro, L. Jamone, and A. Bernardino. Learning intermediate object affordances: Towards the development of a tool concept. In *IEEE International Conference on Development and Learning / Epigenetic Robotics (ICDL/Epirob)*, 2014.

[26] C. Nieto-Granda, J. G. Rogers, A. J. Trevor, and H. I. Christensen. Semantic map partitioning in indoor environments using regional analysis. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1451–1456. IEEE, 2010.

[27] B. Russell and A. N. Whitehead. *Principia mathematica*. Cambridge University Press, 1968.

[28] R. J. Mooney. First order logic (first order predicate calculus. URL `http://www.cs.utexas.edu/~mooney/cs343/slide-handouts/fopc.4.pdf`.

[29] B. R. Duffy and G. Joue. Intelligent robots: The question of embodiment. In *Proc. of the Brain-Machine Workshop*, 2000.

[30] A. KIMMIG, B. DEMOEN, L. DE RAEDT, V. S. COSTA, and R. ROCHA. On the implementation of the probabilistic logic programming language problog. *Theory and Practice of Logic Programming*, 11:235–262, 3 2011. ISSN 1475-3081. doi: 10.1017/S1471068410000566. URL `http://journals.cambridge.org/article_S1471068410000566`.

[31] V. Rodríguez. PCL/OpenNi tutorial 5: 3d object recognition pipeline, 2011. URL `http://robotica.unileon.es/mediawiki/index.php/PCL/OpenNI_tutorial_5:_3D_object_recognition_(pipeline)`.

[32] J. Messias, R. Ventura, P. Lima, J. Sequeira, P. Alvito, C. Marques, and P. Carriço. A Robotic Platform for Edutainment Activities in a Pediatric Hospital. *IEEE Int'l Conf. Autonomous Robot Systems and Competitions (ICARSC)*, 2014.

Appendix A

Base information for the Experiments

A.1 Model of the world

Listing A.1: Map.xml


```xml
<?xml version="1.0"?>
<data>
        <Knowledge>
                <child name="object">
                        <child name="movable_object">
                                <child name="cereal_box"/>
                                <child name="sodacan" />
                        </child >
                        <child name="imovable_object" >
                                <child name="table"/>
                                <child name="shelf"/>
                                <child name="bookcase"/>
                        </child >
                </child >
                <child name="place">
                        <child name="kitchen"/>
                        <child name="living_room"/>
                        <child name="room">
                                <child name="bedroom"/>
                        </child>
                        <child name="closet"/>
                </child >
                <child name="door">
               <child name="kitchen_door"/>
```

```xml
    </child>
</Knowledge>
<Actions>
        <Action name="bring">
                <master name="movable_object"/>
                <slave name="␣"/>
        </Action>
        <Action name="bring">
                <master name="movable_object"/>
                <slave name="imovable_object"/>
        </Action>
        <Action name="bring">
                <master name="movable_object"/>
                <slave name="person"/>
        </Action>
</Actions>
<Objects>
        <object name="counter" class="table">
        <size base_area="200" volume="1520"/>
                <occurrence place="kitchen" x="-1.5" y="2" />

        </object>
        <object name="table" class="table">
        <size base_area="200" volume="1520"/>
                <occurrence place="kitchen" x="-0.5" y="0.7" />

        </object>
        <object name="bench" class="table">
        <size base_area="200" volume="1520"/>
                <occurrence place="living_room" x="2" y="0" />
        </object>
        <object name="coke" clas="sodacan">
        <size base_area="0.66" volume="6.6"/>
                <occurrence place="table" x="0.01" y="0.01" t="1"/>
                <occurrence place="bench" x="2" y="0" t="2" />
                <occurrence place="counter" x="0.5" y="0.75" t="3" />
        </object>
        <object name="cereal" class="cereal_box">
        <size base_area="0.66" volume="16.6"/>
```

```
                <occurrence place="table" x="0.01" y="0.01" t="1"/>
                <occurrence place="bench" x="2" y="0" t="2"/>
                <occurrence place="counter" x="0.5" y="0.75" t="3"/>
            </object>
        </Objects>
</data>
```

A.2 Problog program for the first time-step of object occurrences

```
sodacan(coke).
volume(coke,6.60).
mobility(coke,1).
cereal_box(cereal).
volume(cereal,16.60).
mobility(cereal,1).
1::movable_object(X):-cereal_box(X).
1::movable_object(X):-sodacan(X).
1::object(X):-movable_object(X).
1::imovable_object(X):-table(X).
oclusion(X,Y):-volume(X,VX),volume(Y,VY),VX>VY.
query(is_in(_,_)).
query(oclusion(_,_)).
0.33::is_in(X,counter);0.33::is_in(X,table);0.33::is_in(X,bench):-object(X).
P::seen(X,Y,T1,T2):-is_in(X,Y),mobility(X,M),P is 0.6+0.1*exp(-(T1-T2)/10*M).
P::seen(X,Y,T1,T2):-\+is_in(X,Y),mobility(X,M),P is 0.3-0.1*exp(-(T1-T2)/10*M).
evidence(seen(coke,table,1.0,0.0)).
evidence(seen(cereal,table,1.0,0.0)).
```

Appendix B

Message Topics-commented

B.1 Teaching topic

- `semantic_map/knowledgeInstance[]knowledge` - List of knowledge instances
  - ⋆ string name - Name of the knowledge instance
  - ⋆ string properties - Properties associated with the knowledge instance
- `semantic_map/drelation[]directedRelations` - List of directed relations (ex: Hierachical relations)
  - ⋆ string master - Name of the subject of the action
  - ⋆ string action - Name of the action
  - ⋆ string slave - Name of the object of the action
- `semantic_map/actions[]actionss` - List of actions the robot can perform
  - ⋆ string action - Name of the action
  - ⋆ string instance1 - Name of the subject of the action
  - ⋆ string instance2 - Name of the object of the action
- `semantic_map/Fobject[]objs` - List of objects in the environment
  - ⋆ string name - Name of the object
  - ⋆ string clas - Class of the object (Knowledge instance associated with this object)
  - ⋆ string base_area - Area of the base of the object
  - ⋆ string volume - Volume of the object
  - ⋆ `semantic_map/occurrence[]occurrences` - List of places where the object has been seen
    - ∗ string place - Place where the object was seen
    - ∗ float32 x - X coordinate in the reference frame of the place
    - ∗ float32 y - Y coordinate in the reference frame of the place
    - ∗ float32 time - Time of the observation

B.2 Initial Belief message

- string[] location - List of the names of the locations where the objects can be.
- float32[] coke - Distribution of the probabilities of the coke object over the different placeholders.
- float32[] cereal - Distribution of the probabilities of the cereal object over the different placeholders.