KTH Computer Science and Communication

School of Computer Science and Communication
CVAP - Computational Vision and Active Perception

Topological spatial relations for active visual search

Kristoffer Sjöö, Alper Aydemir, David Schlyter and Patric Jensfelt

TRITA-CSC-CV 2010:2 CVAP 317

Kristoffer Sjöö, Alper Aydemir, David Schlyter and Patric Jensfelt
Topological spatial relations for active visual search

Report number: TRITA-CSC-CV 2010:2 CVAP 317
Publication date: September, 2010
E-mail of author(s): [krsj,aydemir,dschlyter,patric]@kth.se

Reports can be ordered from:

School of Computer Science and Communication (CSC)
Royal Institute of Technology (KTH)
SE-100 44 Stockholm
SWEDEN

telefax: +46 8 790 09 30
http://www.csc.kth.se/

Topological spatial relations for active visual search

Kristoffer Sjöö, Alper Aydemir, David Schlyter and Patric Jensfelt
Centre for Autonomous Systems
Computational Vision and Active Perception Lab
School of Computer Science and Communication
KTH, Stockholm, Sweden
krsj@kth.se

September 15, 2010

Contents

Abstract

If robots are to assume their long anticipated place by humanity's side and help us in our unstructured environments, we believe that adopting human-like cognitive patterns will be valuable. These environments are the products of human preferences, activity and thought; they are imbued with semantic meaning. In this paper we investigate qualitative spatial relations with the aim of both perceiving those semantics, and of using semantics *to* perceive. More specifically, in this paper we introduce general perceptual measures for two common *topological spatial relations*, "on" and "in", that allow a robot to analyze a scene in terms of these relations using real sensory data. We also show how these spatial relations can be used as a way of guiding visual object search. We do this by providing a principled approach for *indirect search* in which the robot can make use of known or assumed spatial relations between objects, significantly increasing the efficiency of search by first looking for an intermediate object that is easier to find. We explain our design, implementation and experimental setup and provide extensive experimental results to back up our thesis.

1 Introduction

While fiction has for a long time painted a picture of robots walking the Earth alongside us humans, in reality robotics has to date mostly been about industrial robots. The industrial robot revolutionized the manufacturing industry with its speed and precision and is still penetrating new markets, although at a somewhat slower pace. Now, we are seeing the beginning of a new era, an era where robots will actually walk, or at least move about, among people. What has been brewing in labs around the world for decades is now very slowly starting to see the light of day. One of this new breed of robots is the service robot, who is intended to help us in the home or office, be it with cleaning, giving us a hand getting up or reminding us to take our medicine. These new robot systems require a completely new level of versatility and adaptability if they are to be able to operate side by side with humans.

There are many important issues that need to be addressed before we have service robots that go beyond vacuum cleaners and floor scrubbers. Some of these issues come from the fact that the robot will be mobile, with all attendant complications such as safety or power supply. There is, however, a whole set of very challenging problems that arise from an unstructured environment where not everything is known in advance, either regarding the properties of the environment or the kinds of tasks to be performed. Humans are superbly adapted to these kinds of conditions; not just physically (such as having legs to negotiate stairs and thresholds, and arms for opening doors and using appliances), but in terms of perception and mental abilities as well.

In this context, being able to perceive and act upon *semantic* information about the environment is key. This entails the ability to go beyond simple, hard-coded decision paths operating on low-level, metric, numerical data. The robot must been endowed with the capability to handle the real world in all its complexity, combining the actual sensory input with "commonsense" knowledge such as the typical location of objects, their function and their relation to other entities. It also needs to be able to communicate

with humans and reason about human concepts in a way that allows it to take context into account when performing tasks.

In this paper we make two important contributions towards this ambitious vision. First, we introduce quantitative measures for two crucial topological spatial relations, which allow the robot to reason about space in a way that comes closer to human cognition. Second, we address the issue of searching for objects and show how to make use of these relations for indirect search in a principled way. Both of these contributions are important additions towards the realization of semantic perception for robots, the former being a means of "perceiving semantics" and the latter of "perceiving, *using* semantics".

1.1 Motivation

For a discussion on how our work relates to prior work in the respective area, please see the dedicated sections. Here we will instead motivate why and how these two issues (spatial relations and search) are important.

The motivation to study topological spatial relations comes from the insight that adopting human-like cognitive patterns is likely to help robots approach human-like performance in the context of homes, offices or other environments that are the products of human preferences, activity and thought. Furthermore, the correspondence between language and cognition means that linguistic concepts may provide an enlightening insight into the nature of those cognitive patterns. Human cognition, language, and civilization have all evolved, and are evolving, in close interaction with each other.

Our research addresses *spatial* concepts specifically. Spatial concepts are of great importance to robotic agents, especially mobile ones, as they:

- are a necessary part of linguistic interaction with human beings, both when interpreting utterances with a spatial content and when formulating such utterances.
- allow knowledge transfer between systems, whether different robots, or databases such as the Open Mind Indoor Common Sense database ([1], which contains "commonsense" information about indoor environments provided by humans, such as where objects may be found), as long as those concepts are shared.
- provide qualitative abstractions that facilitate learning, planning and reasoning. By making use of spatial relations to model a scene in abstract terms the required amount of data will be drastically reduced. In planning and reasoning one can move away from a continuous metric space which is hard to deal with.

Drawing inspiration from results in psycholinguistics, we examine the functional spatial relation of mechanical support, which in English corresponds to the preposition "on", and containment, which corresponds to "in". We contribute novel and general perceptual measures that allows a robot to analyze a scene in terms of these relations in practice using real sensory data. We implement these perceptual models to allow sampling to generate a conditional probability distribution over object poses given the relations that hold for the objects.

Objects play an integral role in how humans perceive and describe space ([2]) and they largely define the functional properties of space. Detecting and recognizing objects will be necessary for robots to be able to function where humans do. The field of computer vision has devoted a considerable amount of effort to these problems but it has largely done so in the context of single image frame where the task is to tell if a certain object instance or class is present or not. While this is far from a solved problem and still holds many challenges we feel that one also needs to address the full robotic version of the problem which includes actually placing the robot and its sensors in a position to detect the objects. One can rarely assume that the object will be in front of the camera and thus the robot, just like us humans, will need to search for objects; a process where recognition is just one of the building blocks. There is a strong connection between object search and the idea of active perception as introduced by [3]. In this paper we will show how one can make use of spatial relations and the idea of indirect search as introduced by [4] to drastically speed up the object search process. The main idea in indirect search is to make use of spatial relations between objects and chain the search, looking first for objects that are easy to find. For example, rather than searching directly for the stapler one first locates the desk, as that is where it is likely to be found. We believe that our use of spatial relations results in the first principled way to realize indirect search.

1.2 Outline

This paper is organized in the following way: Section 2 discusses in more depth prior work on using spatial relations and argues for the need for quantitative measures of relations. Section 3 introduces the spatial relations we are examining and the suggests perceptual models for these. Section 4 introduces the problem of Active Object Search and sets it up as a test case for the use of spatial relations. Section 5 gives an overview of the implementation and some of the design decisions that went into it. Section 6 presents the experimental and Section 7 the experimental results. Section 8 summarizes the work, discusses the results and directions for future research.

2 Spatial Relation Theory

2.1 Prior work

There has been a great deal of investigation into spatial relations within the fields of psycholinguistics and cognitive science; to name a few, [5, 6, 7], but seldom with a view to using them in robotics. There has been work examining ways to quantify spatial relations: [8], inspired by findings on spatial information encoding in the hippocampus, suggests a number of geometrical factors, e.g. coordinate inequalities, that play a part in defining relations such as “below”, “near” or “behind”, but does not attempt to provide exact formulas.

In [9], the *Attention Vector Sum* is proposed as a practical numerical measure of how acceptable a particular spatial relation is for describing a scene, and this model is compared to actual human responses. The scenes used in this work are 2-dimensional and the trajector (mobile object) is treated as a single point.

[10] present a system where a user can sketch images of basic figures, and which learns to distinguish between examples of "in", "on", "above", "below" and "left". However, the domain used in the work is strictly 2-dimensional.

The work of [11] details a computational treatment of spatial relations applied to specific scenes in interaction with a human, and treats the relations in a perceptual framework. The system is demonstrated in simulation, and the focus is on projective relations rather than topological.

Topological relations specifically are surveyed in [12]. *Region connection calculus* (RCC) and its variants provide a language for expressing qualitative, topological relationships between regions, such as containment, tangential contact etc. Relations are of an all-or-nothing nature; and they represent objective and geometrical as opposed to perceptual or functional attributes.

The aforementioned work, because of its emphasis on pure geometry – typically in 2 dimensions – is not directly suited for applications in a practical mobile robotic scenario. This paper, in contrast, takes a novel, functional approach by basing a relation on two fundamental, objective physical properties. Another contribution lies in treating all the objects as entire bodies rather than simplifying them into points, a simplification which ignores the importance of physical contact in the relations. We also show how the method can be used to generate probability distributions.

2.2 Quantitative measures of relations

Spatial relations are sometimes treated as classical logical predicates, with crisp definitions and true/false values. An example is RCC: a region is either a subset of another by the definition, or it is not. Such concepts are necessary for stringent reasoning and proof creation, and highly appropriate for use in formal systems such as computers. However, the world is rarely clear-cut, and human beings are well adapted to this, able to deal with multiple interpretations of a scene and situations that correspond to a greater or a lesser degree to some description ([5, 13]). This gives us robustness to perceptual uncertainty as well as to our categories not corresponding perfectly to reality itself.

In order to achieve similar robustness we evaluate spatial relations not as true/false propositions but via an *applicability function*, a measure of the degree to which a situation is perceived to conform to some ideal conceptualization ([13] terms this an *idealized cognitive model* or ICM). The applicability can be compared with other relations to determine which best describes a given scene, or – given a partial scene – be maximized to yield the configuration that would best match the ideal.

In the following we regard binary relations, between a *trajector* O, i.e. the object whose location is being described, and a *landmark* L which the trajector is compared to. The relation is a function of the form:

$$\textsc{Rel}(O, L) : (\mathbb{R}^3 \times \mathbf{SO}(3)) \times (\mathbb{R}^3 \times \mathbf{SO}(3)) \to \mathbb{R}$$

that is, a mapping from the space of all pose combinations of O and L, to a single value.

2.3 Probability from applicability

Given the above, it is obviously not possible to recover the exact pose of the trajector from the value of the relation alone. However, a probability distribution over poses can be produced in the following way:

Given the geometry of the landmark L, and of the trajector O, each possible pose combination π for O and L yields a value of $\textsc{Rel}(O_\pi, L_\pi)$ for that pose. Introduce a true/false event $Rel(O, L)$, which signifies e.g. that a human describes the configuration using the relation in question. By making the assumption

$$\mathbf{p}(Rel(O, L)) \propto \textsc{Rel}(O, L)$$

extracting a probability density becomes simply a matter of normalising Rel over the space of configurations:

$$\begin{aligned} \mathbf{p}(\pi | Rel(O, L)) &= \frac{\mathbf{p}(Rel(O, L)|\pi)\mathbf{p}(\pi)}{\mathbf{p}(Rel(O, L))} \\ &= \frac{\textsc{Rel}(O_\pi, L_\pi)\mathbf{p}(\pi)}{\alpha} \end{aligned} \tag{1}$$

where α is a normalizing factor. Though it may be hard to express this distribution analytically, by drawing samples randomly from $\textsc{Rel}(O_\pi, L_\pi)\mathbf{p}(\pi)$, and normalising over the result an arbitrarily good approximation can be found.

3 Topological spatial relations: on and in

3.1 Topological relations

Spatial predicates in language come in different categories. *Projective* spatial relations constrain the trajector's location within an essentially *directed* region relative to the landmark. The direction may depend on many factors, such as intrinsic properties of either object, or the frame of reference of an onlooker. Examples in English include "to the left of", "behind" and "past". *Topological* relations, in contrast, locate the trajector in some manner that is independent of direction. Typical examples are "on", "in", "at" and "inside". Topological relations seem to be among the first to be learned in humans ([14]).

Research suggests that verbal descriptions of space do not, in general, correspond one-to-one to cognitive representations ([15]). Instead, it seems conceptualization forms around kernels of *functional* criteria, such as "physical attachment", "superposition" (an object being located in the space vertically above another) or "containment" (an object being enclosed in another).

Topological spatial relations naturally form *hierarchies*, where each object has a single topological "parent" which it is placed in relation to: "It's in the basket on my desk on the sixth floor in that building." Although there are often exceptions, this general tendency nevertheless is very useful for organising spatial information. Topological relations also often exhibit *location control*, i.e. the location of the trajector is physically constrained by the motion of the landmark. This makes the relation stable over time even during dynamical processes, unlike most projective relations.

In the sequel, we look at the two most important topological linguistic descriptions in English, "on" and "in" and their fundamental, functional connotations.

3.2 On

As has been noted by e.g. [16] and [5], English' "on" carries a central meaning also represented in many other languages: that of *support* against gravity; i.e., a trajector is "on" a landmark if it would, were the landmark to be removed, begin to fall or move under the influence of gravity. This sense of "on" is the idealized cognitive model or ICM around which other, less central and more idiomatic senses of "on" form in ways specific to each language.

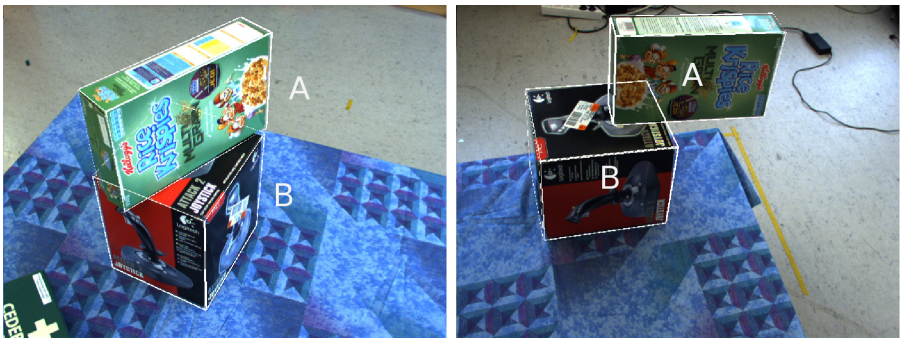
We observe that the notion of support is highly related to the functional aspects of space as designed, constructed and lived in by human beings. Such space is full of entities specifically made to support others, both statically – such as tables, shelves, counters, chairs, hooks and desks – and dynamically – such as trays, trolleys, and dishes.

It thus is of interest to robotics to use a spatial representation that encodes this functional relationship between objects. Although this work is inspired by linguistic clues, giving a robot additional linguistic capabilities is only an incidental outcome. It is also necessary to point out that the word "on" spans far more meanings than the core physical support relation: it may entail indirect rather than direct support, adhesive or suspended support, as well as metaphorical uses. Here, we are not attempting to cover all of that complexity.

3.2.1 A perceptual model

The "support" relation proposed above constitutes an idealized model, but is as such not possible to evaluate directly from perceptual data. Neither robots nor humans can ascertain degree of mechanical support merely by visually regarding a scene, and so it becomes necessary to introduce a perceptual model to estimate the ideal relation. Figures 1(a) and 1(b) illustrate how a human observer's perceptual model detects an anomalous support relation, even though in reality the scene is stable (because of a hidden weight inside A).

Humans use context, experience with specific objects and generalizations, as well as schemata to decide whether an object is "on" another. For robots, we model this by a simplified geometric predicate, termed On, such that $\textsc{On}(A, B)$ corresponds to "A is supported by B". The relation is graded and can attain values in the range $[0, 1]$.



(a) Normal-seeming support relation (b) Abnormal-seeming support relation

Figure 1: Two example scenes with typical and atypical On situations illustrating the perceptual challenge.

The following are our criteria and their justification. O denotes the trajector object, and S the support object or landmark. The criteria are illustrated in Figure 2.

1. *Separation between objects*, d. d can be positive or negative, negative values meaning that objects appear to be interpenetrating.

   In order for an object to mechanically support another, they must be in contact. Due to imperfect visual input and other errors, however, contact may be difficult to ascertain precisely. Hence, the apparent separation is used as a penalty.

2. *Horizontal distance between COM and contact*, l. It is well known that a body O is statically stable if its center of mass (COM) is above its area of contact with another object S; the latter object can then take up the full weight of the former. Conversely, the greater the horizontal distance between the COM and the contact, the less of the weight S can account for, as the torque gravity imposes on O increases, and this torque must be countered by contact with some other object.

   Thus we impose a penalty on $\textsc{On}(O, S)$ that increases with the horizontal distance from the contact to the COM of O. The contact is taken to be that portion of S's surface that is within a threshold, δ, of O, in order to deal with the uncertainties described above. If $d > \delta$, the point on S closest to O is used instead; otherwise, l is the positive distance to the outer edge of the contact area if the COM is outside it, and the negative distance if it's inside.

3. *Inclination of normal force*, θ – the angle between the normal of the contact between O and S on the one hand, and the vertical on the other. The reason for including this is that *mutatis mutandis*, the normal force decreases as the cosine of θ, meaning the weight of O must be either supported by another object or by friction (or adhesion).

All these values can be computed from visual perception in principle. Unless otherwise known in advance, the position of the COM is taken as the geometrical centroid of

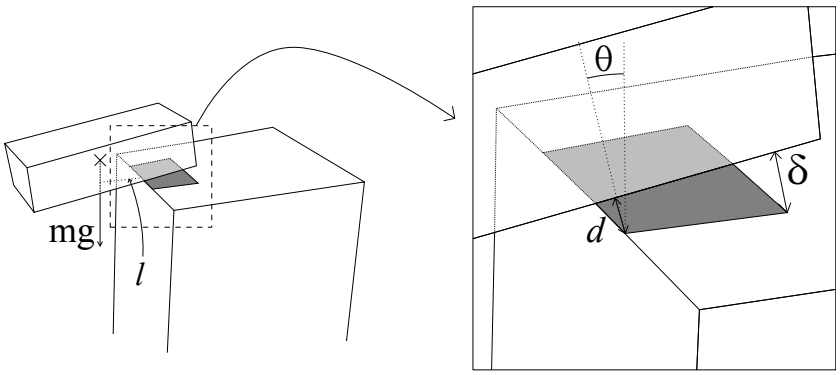


Figure 2: Key features used in computation of On: Separation d, COM offset l, contact angle θ and contact threshold δ. The gray area represents the contact.

the object (since density cannot be determined by vision, compare Fig. 1).

In order to allow a measurable value to be computed, the agreement with each of the three above criteria is represented as a continuous function, with a maximum at the point of best agreement with the criterion. This provides robustness against error. Criterion 1 is represented by an exponential *distance factor*:

$$\text{On}_{distance}(O,S) \triangleq \exp\left(-\frac{d}{d_0(d)}\ln 2\right) \tag{2}$$

where d_0 is the falloff distance at which On drops by half.

$$d_0 = \begin{cases} -d_0^-, & d < 0 \\ d_0^+, & d >= 0 \end{cases}$$

The constants d_0^- and d_0^+ are both greater than 0 and can have different values (representing the penetrating and nonpenetrating cases, respectively).

Criteria 2 and 3 make up the sigmoid-shaped *contact factor*:

$$\text{On}_{contact}(O,S) \triangleq \cos\theta \cdot \frac{1+\exp(-(1-b))}{1+\exp\left(-\left(\frac{-l}{l_{\max}}-b\right)\right)} \tag{3}$$

Here, $l_{\max}$ is the maximum possible distance an internal point can have within the contact area, and b is an offset parameter.

The values are combined by choosing whichever factor is smaller, indicating the greater violation of the conditions for support:

$$\text{On}(O,S) \triangleq \min(\text{On}_{contact}, \text{On}_{distance}) \tag{4}$$

Note that this version of the "on" relation is not transitive. One can introduce a transitive version which is fulfilled by either the above criteria, *or* by O being "on" x which is "on" S.

Given a good enough model of the geometry of a scene, it is possible for a robot to evaluate which objects in its view are "on" which other objects by these definitions. Note again that the idea is to approximate the "ideal" conceptualization of functional support.

3.3 In

Even more prevalent than "on", the preposition "in" has a wide use as a marker of membership in addition to its purely spatial connotations. Here, however, it is on the fundamental cognitive property of *containment* that we shall concentrate.

Man-made environments are replete with entities that are thought of as containers, from entire buildings through cabinets, crates and closets to cups, tins and boxes. Containment entails a wider array of functional relationships than the support concept referred to in the previous section. Apart from location control, there are for example separation, concealment, confinement and protection. These connotations are of great relevance to humans interacting with the objects and an important reason why the concept of containment is needed in cognitive robotics. Yet, each of these functional aspects of containment presupposes some situation to be meaningful. It is beyond the scope of this work to treat such complex and richly contextual relationships. However, we may still successfully regard a perceptual model that encompasses many of the connotations in practice. For this we choose a model representing *enclosure*.

3.3.1 Perceptual model

Enclosure refers to the geometrical subsumption of one object by the convex hull of another. This covers many of the everyday uses of the word, such as "in the house", "in a forest" or "in this bowl"; see [5]. (There are also many exceptions.)

The primary indicator of enclosure is the ratio of contained volume to total volume:

$$\textsc{In}_{enc} \triangleq \frac{V_{O \cap C^{conv}}}{V_O} \tag{5}$$

where V_O is the volume of the trajector object O, and $V_{O \cap C^{conv}}$ the volume of the part of O that falls inside the convex hull of the container object C.

However, if $\textsc{In}_{enc}$ were the only factor determining degree of containment, cases where O and C overlap in space, which is not physically plausible, would evaluate the same as realistic configurations. Because such cases are bad examples of the relation, we supplement the model with a penalty function on perceived object interpenetration:

$$\textsc{In}_{pen} \triangleq \begin{cases} 1 & d \geq 0 \\ e^{d/k} & d < 0 \end{cases} \tag{6}$$

where d is the minimum distance between O and C (as defined in Sec. 3.2.1) and k a falloff constant.

The total applicability function for the containment spatial relation is taken to be:

$$\text{IN} \triangleq \text{IN}_{enc} \cdot \text{IN}_{pen} \tag{7}$$

Note that, unlike the case with the support relation in Sec. 3.2.1, this definition ignores gravity. It is easy to find cases where the use of the word "in" is affected by gravity, but its importance is less crucial than for "on".

4 Putting spatial relations to use: AVS

4.1 Background

The ability to find 3D objects in a 3D world is an important item on a mobile robot's skill repertoire. Visual search entails ascertaining the location of a specific object, or of one or more of a given class of objects, and doing so in an efficient manner. The existing body of work on active visual search by a mobile robot is not extensive, however. Previous work is mostly based on the assumption that the object in question is within the field of view of robot's sensors ([17, 18, 19, 20, 21]); in the case of mobile robots, on the other hand, the exact location of the target object is assumed to be known. The pursuit/evasion literature provides some hints for the problem by providing methods to plan a search on a graph representation. However the high level graph representation often neglects lower level details of active visual search that are of great import, such as occlusion and the field of view of the sensors.

An often-stated reason by researchers in the field for assuming that the object is in robot's immediate sensory scope, is that tasks such as object recognition and manipulation already pose hard enough challenges. However as we crack these challenges, the assumption that the object of interest is in the field of view of the robot must be abandoned. A robot tasked with a fetch-and-carry task is unlikely to have the target object in its immediate reach or know the location of every object in a home environment. In line with this reasoning, recently visual search has received attention from researchers as robots interacting with objects become more and more a reality ([22, 23, 24, 25]).

Still, searching for an object uninformed, i.e. without any prior information on its location, is simply not tractable given the limited field of view of typical sensors and the large search space that needs to be covered. Despite a series of fruitful contributions to the active visual search problem, including algorithms for covering a known or previously unknown world efficiently ([26, 27, 28, 29, 30]), the object search problem is shown to be NP-hard by [31]. Therefore we can only hope to find a solution by approximation of the optimal search behavior. In the case of a mobile robot operating in an everyday environment, exploiting the semantics of the environment provides strong cues for such approximations. Therefore a principled approach to searching for objects by exploiting a priori semantic information is much needed. Ideally a robot with a specific task of locating an object should make use of all the bits and pieces of evidence; be it from an overheard dialogue, a target object's class limiting the search to a specific region (e.g. forks are usually found in the kitchen) or a known spatial relation between the target and some other entity.

It is well known that human vision is greatly helped by available context information ([32, 33]). Often the context is acquired through examining other objects and shapes that are present. One powerful idea which naturally involves integration of multiple cues is *indirect search*. Indirect search is about first looking for an intermediate object in order to find the target object by exploiting the relation between the former and the latter. This can be exemplified by first searching for the larger and easier-to-detect whiteboard, and then looking for the pen next to it. It involves fusing multiple types of cues, which is difficult and not yet in place in the previous work. In this paper we provide a principled way of performing indirect search in a 3D environment throught the use of spatial relations.

The goal of the active visual search process performed by a mobile robot is to calculate a set of sensing actions which brings the target object, in whole or partly, into the sensor field of view so as to maximize the target object detection probability and minimize the cost.

4.2 Problem formulation

Here we briefly describe the active visual search problem following the formulation of [30]. Let Ψ be the $2D$ search region whose 2D structure is known *a priori*. To discretize the search region, Ψ is tessellated into identically sized cells, $c_1 ... c_n$. The area outside of the search region is represented by a single cell c_0. A sensing action s is then defined as taking an image of Ψ from a view point v and running a recognition algorithm to determine whether the target object o is present or not. In the general case, the parameter set of s consists of camera position (x_c, y_c, z_c), pan-tilt angles (p, t), focal length f and a recognition algorithm a; $s = s(x_c, y_c, z_c, p, t, a)$. The cost of a search plan $S = s_0 ... s_i$ is then given as $C(S)$.

An agent starts out with an initial probability distribution (PDF) for the target object's location over Ψ. We assume that there is exactly one target object in the environment either inside or outside the search region. This means that all cells will be dependent and every sensing action will influence the values of all cells. Let β be a successful detection event and α_i the event that the center of o is at c_i. The probability update rule after each s with a non-detection result is then:

$$\mathbf{p}(\alpha_i|\neg\beta) = \frac{\mathbf{p}(\alpha_i)(1 - \mathbf{p}(\beta|\alpha_i))}{\mathbf{p}(\alpha_0) + \sum_{j=1}^{n} \mathbf{p}(\alpha_j)(1 - \mathbf{p}(\beta|\alpha_j))} \tag{8}$$

Note that for $i = 0$, $\mathbf{p}(\beta|\alpha_i) = 0$, i.e. we cannot make a successful detection if the object is outside the search region . Therefore after each sensing action with a non-detection result the probability mass inside Ψ shifts towards c_0 and the rest of Ψ which was not in field of view.

4.3 Next best view selection

The next step is to define how to select the best next view given a PDF. First, candidate robot positions are generated by randomly picking samples from the traversable portion

of Ψ. This results in several candidate robot poses each with associated view cones. We define a view cone as the region of the space which lies in the field of view of robot's camera. For a given camera, the length of the view cone is given by the greatest distance at which the object can reliably be detected, which depends on the size of the object.

The next best view point is then defined as:

$$\underset{j=1..N}{\operatorname{argmax}} \sum_{i=1}^{n} \mathbf{p}(c_i) V(c_i, j) \tag{9}$$

Where N is the number of candidate view points and V is defined as:

$$V = \begin{cases} 1, & \text{if } c_i \text{ is inside of the sensing volume of the } j^{\text{th}} \text{ sensing action} \\ 0, & \text{otherwise} \end{cases}$$

The greedy approach followed here aims to prioritize regions of the environment where the probability of containing the target object is highest. Also note that the factor that influences the search the most is determining and updating the object probability distribution.

4.4 Spatial relations and AVS

Assume that some algorithm exists that produces a sequence of views, given a probability distribution for the sought object $p(\pi_O = x) = f_O(x)$, the views incurring the total cost $C_O\{f_O\}$. The cost may depend on the actual object, due to size, saliency et cetera.

In this context, topological relations can be highly useful. In many scenarios, the exact position of an object O may be uncertain or unknown, even while it is known or presumable that it is e.g. On some other object S. This information can have several sources: O may have been seen On S at an earlier time, and location control implies the relation will still hold even if S has moved. The connection may also be statistical in nature, learned through experience from many analysed scenes ("this type of object is usually located On that type") or from a commonsense knowledge database. The information may also come from symbolic reasoning or linguistical utterances.

Using an object's location to help search for another is known as *indirect search*. Indirect search was first investigated by [4]; there, a system looking for a phone in a room is first tasked with finding the table that the phone is resting on. [29] re-visited the idea of indirect search in the context of mobile robotics; however, previous work on exploiting spatial relations to guide the visual search process on mobile robots is non-existant.

The principle we propose for indirect search is the following: given a target object (e.g. "book"), and a topological relation hierarchy (e.g. "book In box On desk"), we begin by searching for the "base" object, and once that has been found we compute a posterior PDF for the next higher object and search for it, etc. Details are provided in the following section.

5 Implementation

5.1 Representation

The principles set out in the preceding sections are of an idealized nature and not immediately amenable to practical application. In this section we explain the salient points of our software implementation of the theory and the further considerations and assumptions necessary.

5.1.1 Grid map

The intrinsically three-dimensional nature of both our proposed spatial relation concepts and the visual search task itself means that the way space is represented is crucial. There are many different possibilities commonly used in different applications. *Voxel* representations use a 3-dimensional Cartesian grid, usually of uniform resolution along each axis. They are conceptually simple and easy to implement, but can be costly in terms of memory and processing requirements. *Tree* representations, including KD-trees and octrees ([34]), tackle the problems of voxel grids by dynamically dividing up space into smaller subcomponents as needed. These representations are efficient for storage, but less suited for dynamic modification.

We have chosen in this work to use a two-and-a-half-dimension representation, which attempts to combine the advantages of a fixed-resolution grid with those of dynamic resolution representations. It is similar to the system used in [35]. The choice is based on the assumption that the way space is actually organised within an indoor environment is different in the vertical and the horizontal. Typically, within a room or one floor of a building, the number of distinct objects superimposed at the same horizontal location is limited. We therefore introduce a dynamical aspect in the vertical dimension only.

The horizontal plane is divided up into a Cartesian grid. Each cell in the grid is associated with a *column*, representing the extrusion of that cell along the z-axis.

Each column is subdivided into an arbitrary number of segments, which we term *bloxels*. In other words, bloxels resemble voxels in that they are box-shaped, generally small, and regularly spaced along the x and y axes. They differ in their dynamical extent along the z-axis.

Each column can have a different bloxel subdivision. No bloxels overlap, and they together exhaustively cover the column from the minimum to the maximum vertical coordinate defined for the map. Bloxels have a given minimum size; bloxels smaller than this are absorbed within their neighbors.

Each bloxel is associated with two values: *occupancy* and *probability density*. The former is a three-state variable that can take on the values OCCUPIED, FREE and UNKNOWN. Occupancy is used when evaluating possible view cones and when performing probability updates (see below).

The probability density corresponds to the current estimate of the probability of the currently sought object being located at any point within the bloxel. The probability mass associated with each bloxel is equal to its PDF value multiplied by its size along the z-axis.

The grid map is initialized with one bloxel for each 2D position, which extends from the minimum to the maximum z-value, with values UNKNOWN / 0.0. We split this bloxel when we have acquired information and merge bloxels with similar values along the vertical – similar in spirit to what is done in for example Octomap ([36]) – to compact the tree.

5.1.2 View cone evaluation

The main task of the visual search algorithm is figuring out where to point the camera next. In the case of a tilt-camera mounted at a fixed height, this entails determining four parameters: camera position in x and y; camera pan, and camera tilt. To make the search faster, we treat it in two steps. The x, y and pan parameters are determined using a random sampling approach. Free space in the map (i.e. not UNKNOWN or OCCUPIED) is sampled randomly together with a random bearing, yielding a set of 2-D candidate views. Each 2-D view is evaluated on the basis of the total probability mass of all columns (all bloxels with the same x,y-position) that it encompasses. This constitutes an optimistic estimate of the highest probability mass of any one 3-dimensional view cone with those same parameters.

To select an actual 3-D view, we pick out the best 5% of the 2-D views, and sample 5 tilt angles for each. The resulting 3-D candidate view cones are overlaid on the bloxel map, and the probability mass extracted. The highest massing cone is selected as the next view.

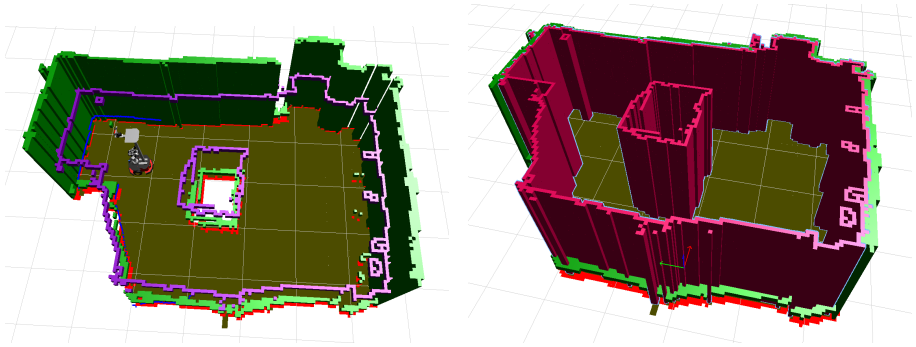
When evaluating a 3-D view cone, only bloxels that are within a certain range of the camera are considered. This range is based on the size of the object: $r_{\min} = l/\left(2\tan\left(\frac{1}{2}\alpha_{\min}\right)\right)$ where l is the largest dimension of the object, and α_{min} is the smaller of the horizontal and vertical view angles. This formula picks a minimum distance to an object such that the object will fit within the camera image. Any closer and the chance of successful detection drops sharply. A maximum range is selected as a multiple of the minimum range: $r_{max} = k \cdot r_{min}$. The proper setting of k depends on the sensitivity of the detection algorithm to increasing distance. We use the value $k = 4$ in this work. r_{max} is capped at 5 m for reasons of performance of the view planning.

If any obstacles lie within the view cone, the portions beyond them are occluded and do not count when evaluating a view cone's covered probability mass, nor are they changed by view updates.

5.1.3 Heuristic priors

For the cases when an object's pose cannot be derived from another, such as for the base object in a hierarchy, or when no relational information is available, an uninformed prior must be used. A completely uniform prior over all space is the most obvious option. However, this makes no use at all of whatever sensory information is available and leads to highly inefficient searches.

If a dense obstacle point cloud were available, these might be profitably used to create priors – see [37]. In our case, however, we only have access to data in the form of 2-D



(a) Prior distribution, given height (b) Prior distribution, unknown height

Figure 3: Prior PDFs. The purple cells represent the non-zero region.

laser scans, and so can only know about obstacles at a fixed z height. Still, things do not hang suspended in mid-air, and most objects will – directly or indirectly – be supported by some contact with the ground. Therefore, it is reasonable to take obstacles at ground level as a cue to the potential of objects existing above them. Similar approaches have been used in e.g. [38, 39].

Accordingly, uninformed priors heuristically ascribe probability mass to map columns that have laser obstacles as neighbours. Some objects, such as pieces of furniture, can be assumed to be standing on the floor, and thus to have a known z-coordinate, meaning the prior is concentrated at that height. Other objects have entirely unknown poses and their priors are consequently assigned over the entire height range of the column. Fig. 3(a) shows an example where the object is located at a known height (all probability mass is concentrated at that height in regions where the laser has detected objects). Fig. 3(b) shows an example of a case where no height information is provided.

5.1.4 Point cloud sampling

In Section 2.2, it was explained how functions like On and In implicitly define probability densities over configuration space. In practice, though, it is necessary to sample them, as the functions are not analytical and as our map representation is discrete (Section 5.1.1).

For the case when a landmark L has a known pose, the space to sample is the space of poses of the trajector O, which has 6 degrees of freedom. We sample this space using a 3-dimensional grid centered on the landmark for the position of O, and a uniform partition of $\mathbf{SO}(3)$ for its orientation – i.e. the sampling is systematic.

To populate the grid map we then perform kernel density estimation (KDE) with a triweight kernel:

$$K(u) = (1 - u^2)^3 \quad |u| \leq 1$$

placed at each sample point and weighted by the value of the relation there. The result is then normalised. (An example of the procedure can be seen in Figs. 9(d) and 9(e).)

When the pose of L is also unknown, but a prior probability for the position is given, the space to be sampled becomes 12-dimensional in principle and the cost becomes quite high. We ameliorate this problem by caching sample point clouds so that they need not be recomputed each time. We also make use of the fact that the relations' values are invariant to translating both objects by equal amounts, as well as of the relative sparseness of our priors (see Figure 3).

Sometimes a distribution is desired for an object that is more than one step removed from the prior in the topological hierarchy, e.g. "book In box On table". We can then compose together the results from sampling "book In box" and "box On table", before performing the KDE, by convolving them in the spatial dimension and summing out the intermediate object (the box, here). This allows for performing visual search using topological relational information without necessarily performing indirect search (see 6.3).

5.2 Object detection

Once a view has been selected and the robot has moved to the designated location and turned the camera in the designated direction, a monocular image is captured. SIFT recognition is then run on the image for each of the objects in the database.

On detecting an object, the pose of the object is estimated from the matching key points using the system described in ([40]), and the object's model is put in the grid map as an obstacle, so that it will occlude future views properly.

If the object which is currently being searched for (the target object, except in the case of indirect search, where it may be a container or support object) is not detected in a view, a probability update is performed on the space encompassed by that view cone, less any parts that are occluded. This reduces the posterior probability density within the cone, and increases it elsewhere in the map, while the total probability that the object is in the room decreases (Sec. 4.2).

5.3 Visual search algorithm

The following schematically summarizes the procedure used by the robot to find a target object:

1. Select current object.
    - If indirect search mode, select base object in hierarchy.
    - Otherwise, select the target object.
2. Generate a prior for the current object.
    - If the current object is In or On another object with known pose, use KDE to create a prior at that location. (5.1.4)
    - If In or On another object with unknown pose, use KDE around a sampled set of possible locations (where there are obstacles).

- Otherwise, use a prior based on obstacles. (5.1.3)

3. Sample view cones randomly in accessible space. (5.1.2)
4. Go to the best view point and perform object detection. (5.2)
   - The detector is run for all objects, not just the current.
5. Insert any objects detected into the map.
6. If the current object was not detected, adjust the probability density inside the view cone accordingly. (4.2)
7. For indirect search, check if any object higher in the hierarchy than the current has been detected. If so, make that the current object, and repeat from step 2.
8. If too many views have been processed already, or the posterior probability that the object is actually not in the room at all ($\mathbf{p}(\alpha_0)$) is too high (see Sec. 4.2), terminate.
9. If the target object is not yet detected, repeat from step 3.

6 Experimental setup

6.1 Hardware setup

The robot used in our experiments is a Pioneer III wheeled robot, equipped with a Hokuyo URG laser range finder and a stereo camera (with no zoom capability) mounted on a pan-tilt unit at 1.4 m above the ground (see Fig. 4). The system uses a SLAM implementation ([41]) for localization and mapping and builds an occupancy gridmap based on laser data. A map was prepared in advance and used in all experiments so that they would have the same preconditions.

The experiments were carried out in a room with dimensions 6 m $\times$ 5 m furnished as a living room, with two couches, a low table, a desk, and three bookcases – two large and one small. Figure 6 shows the different objects used.

6.2 Experimental layouts

In every experiment the robot was tasked with locating a specific book (see Fig. 6(a)). The qualitative location of the book was one of six alternatives:

1. In the box
2. On the table
3. In the box, on the table
4. In the small bookcase

Figure 4: The robot used in the experiments

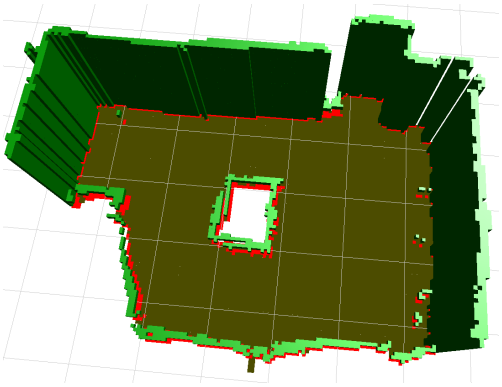
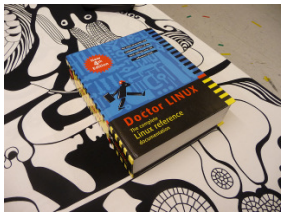


Figure 5: The initial map provided the robot



(a) Book

(b) Table

(c) Box

(d) Small bookcase

(e) Large bookcase

Figure 6: Detection objects

5. On the small bookcase

6. In the large bookcase

6.2.1 Scene setup

As the spatial relations listed above were intended to constitute the qualitative data that was to be provided to the robot, scenes needed to be set up that agreed with each respective description. In order to minimize the experimental bias in the positioning of the objects, we asked 10 individuals to set the objects up according to the following script:

> Now we will ask you to perform a few tasks involving moving some objects, and between the tasks we will take a picture of the result.
>
> 1. Put this box on top of either of the couches and then put the book in the box.
> 2. Put the book on top of the table.
> 3. Put this box on top of the table and then put the book in the box.
> 4. Put the book in that bookcase. [Indicating the smaller bookcase]
> 5. Put the book on top of that bookcase. [Indicating the smaller bookcase]
> 6. Put the book in that bookcase. [Indicating the larger bookcase]

(Note that layout #1 corresponds to an unknown location for the box, as far as the robot is concerned.)

The gestures used to indicate the objects and pieces of furniture were kept as sweeping as possible so as to remove any influence on the precise positioning of the objects. In all but a few instances, the subjects placed the objects without any further exchange. On some occasions a subject asked for confirmation or further feedback; this was restricted to "Anywhere is fine" or silent nods to the same effect.

The subjects were 10 in number, of which 6 were male and 4 female. All were fellow researchers or students, not connected with the present work, and all were proficient in English although there were no native speakers. Figure 7 illustrates some of the resulting object layouts.

The results were in many respects similar across subjects, with some predictable tendencies: All subjects placed the cardboard box with the opening facing upward. Nearly all laid the book down on the table and on top of the bookcase, while they in contrast stood it up inside the bookcases. No one placed the book standing up inside the box. These observations, though anecdotal, point at the importance of functional aspects, as well as schemata, to spatial language and cognition: books are "supposed" to stand up in bookcases (as this makes retrieval easier), but no such tendency affects the other situations, where it is instead general stability that wins out. Similarly, the role of the box as container is typically one of location control, and so the configuration most suitable for this purpose is chosen.

(a) "In the box, on the table" (b) "In the large bookcase" (c) "On the small bookcase"

Figure 7: Three example object layouts

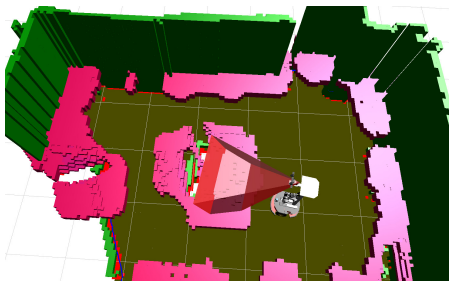


Figure 8: Informed prior: "book On table"

6.3 Search modes

For each qualitative object setup, three experimental runs were performed:

Uninformed search (U): The first mode ignores spatial relation information entirely, using only the obstacle- based heuristic prior described in Sec. 5.1.3. Given no additional information, the robot assumes that the book might be located at any point in the map where there is an obstacle (as provided by the laser scanner), and at any point between floor and ceiling (compare Fig. 3(b)).

Informed direct search (D): In the second mode, the robot also searches directly for the book; however, it uses a prior based on the given qualitative spatial information. For example, with the book "in" the box and the box "on" the table, it utilises the information that the table will be located at a given height above the floor, as well as the cumulative uncertainty of the book's position within the box and of the box' on the table. Figure 8 shows an example of the prior obtained from "book on table". Note that we do not actually look for the intermediate objects here, they are simply used to narrow down the search space. The height of the table is known, and thus we know, approximately, at what height to look for the book, but not the x,y-position.

Indirect search (I): The final search mode searches for objects in the order given by their spatial hierarchy: a support before its supported object, and a container before its contents. The justification for this is that containers and supports tend to be larger and thus more easily detectable by the robot, allowing it to cover the room in fewer views. Containers, furthermore, often obscure their contents and a good pose estimate of the container may be crucial in selecting a good view angle for acquiring the contents.

Although views were selected for the next object in the hierarchy, detection was executed for each object at all times. If the target object was ever detected prematurely the search would terminate successfully. Similarly, if in the "book in box on table" setup the box is found before the table, search moves immediately into the final search phase using the box' location.

6.4 Initial knowledge

To begin with, the robot is provided with a database of the objects and their appearance. It does not know their pose in the room, although the table and bookcases are restricted to be in an upright position and standing on the floor.

The robot is also given a map of the room, as recorded in a previous run using the laser scanner. The occupied cells that correspond to actual walls have been manually labelled, and for these cells the whole column is marked OCCUPIED from floor to ceiling; for the rest, only the portion at the height of the laser scanner is OCCUPIED while the rest is unknown. Figure 5 shows the initial map. Note that laser data is not used for object detection. Finally, the true[1] object relations were given to the robot.

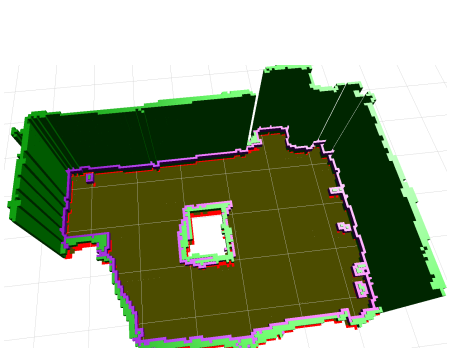
These were the parameters used during experiments; see previous sections for details.

Symbol	Parameter	Value used	Reference
d_0^+	Contact penalty, outside	0.035 m	3.2.1
d_0^-, k	Contact penalty, inside	0.023 m	3.2.1
δ	Patch threshold	0.030 m	3.2.1
b	Center-of-mass offset	-0.05	3.2.1
	Percentage 3D views evaluated	5%	3.2.1
	# of sampled 3D views per 2D view	5	3.2.1
$p(\beta\|\alpha_i)$	Sensor model (detection likelihood)	0.8	4.2
$p(\alpha_0)$	Initial prob. object is not in room	0.3	4.2
	# of sampled 2D views	100	5.1.2
	Max views	15	5.3
	Max $p(\alpha_0)$	0.7	5.3

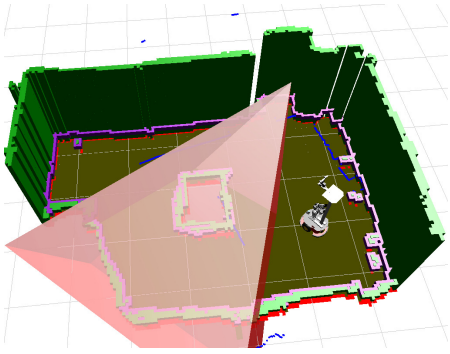
6.5 Example run

An example of a successful indirect search run is shown in Figure 9. Looking for the book, and given that "book ON table", the robot first searches for the table, by selecting

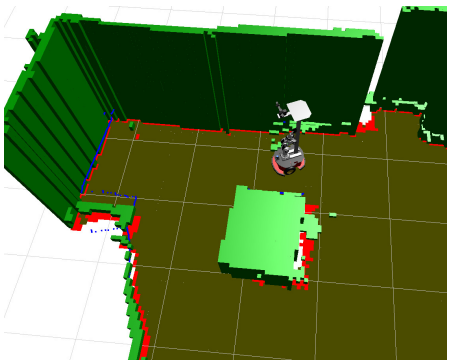
[1] As conceived by the test subjects



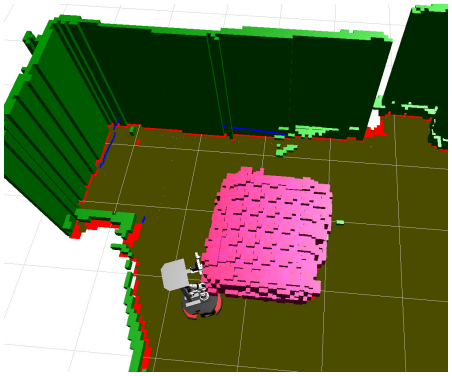
(a) Initial map (green), with prior for table (purple)



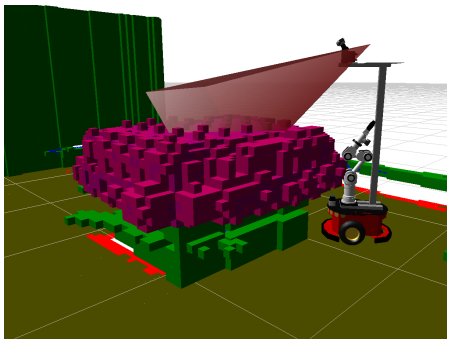
(b) A first view is selected



(c) The table is detected and inserted into the map



(d) PDF for "book On table"



(e) Next best view selected

(f) Book detected

Figure 9: A successful indirect object search procedure

a view cone that covers as much as possible of the prior distribution. In this case, it detects the table successfully, and projects the model of the table into the map in order to model its occlusion properly. Given the pose of the table, samples are acquired for the On function and KDE is used to populate the region with probability mass. Again, a view cone – this time with a range corresponding to the book – is selected so as to cover the maximum probability mass. With this view, the book is located successfully.

7 Results

For each of the 10 subjects and each of the object layouts set up by those subjects, the system was run once using each of the three strategies outlined above. The exception was that type "D" search was not carried out for layout #1[2]. In total, thus, 170 runs were performed, each taking between 2 and 10 minutes.

7.1 Reliability

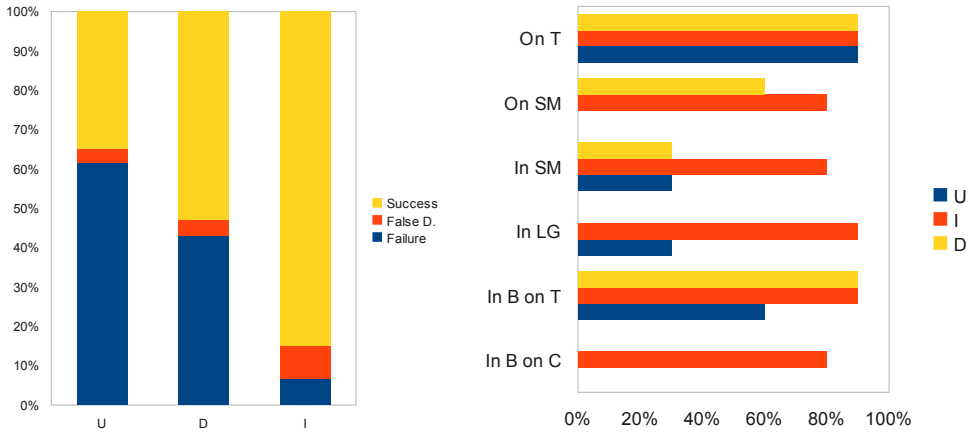
Figure 10(a) summarizes the outcomes across all the subjects and locations. It is very clear that indirect search performs considerably better in these tests in terms of actually locating the object within the allotted number of views. This is due to three factors:

- The relational information restricts the possible positions of the trajector, reducing the space that needs to be searched.
- The fact that the table, box, and bookcases are all larger than the book means that detection can take place at a greater range, and that the robot can place itself so as to capture a larger portion of the room with each view.
- Detecting the container before looking for the trajector means that the robot can take into account the occlusion that the container imposes, and select views that are not blocked by e.g. the sides of a bookcase.

In contrast, direct informed search can leverage only the first of the above. Still, the reduced search space suffices to make a difference in these tests. It should also be noted that the advantages of indirect search are dependent on the reliable detection of the support or container objects; if these are hard to see, direct informed search may outperform indirect search.

Figure 10(b) shows the success rate as a function of the location of the trajector. Although the data is not extensive enough for far-reaching conclusions, a trend can be discerned: direct informed search is most helped by the On relation. This might be expected, as that relation places a sharp restriction on the z-coordinate of the trajector, which is not the case with In. (Note that no experiments were performed with direct informed search for "In box On couch".)

[2]This is because there was not yet any way of incorporating a prior without a fixed z-coordinate in the chained sampling algorithm. As "In box" provides next to no information when the orientation is unknown, results for these tests would be expected to be no better than uninformed search.



(a) Overall success rate

(b) Success rate by location

Figure 10: Success rate for all experiments (B = Box, SM = Small bookcase, LG = Large bookcase, T = table)

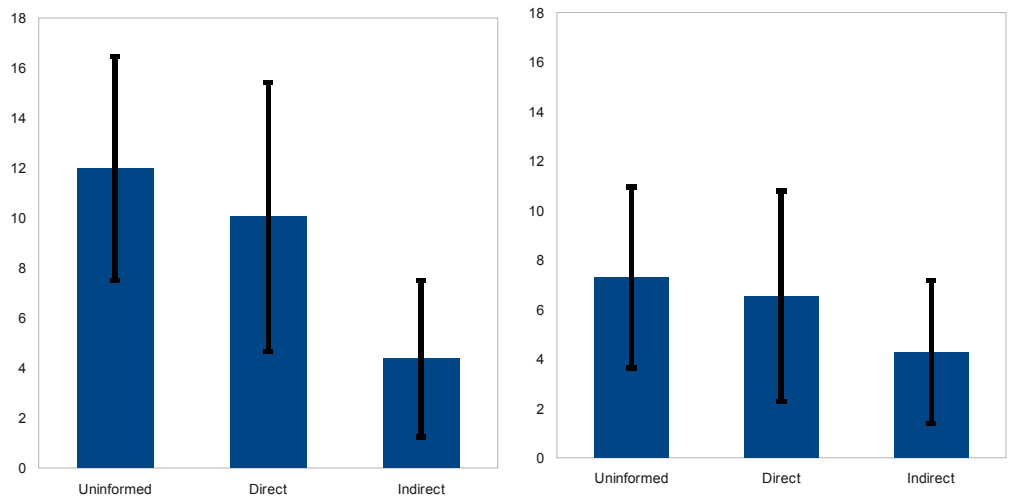
Indirect search does exhibit a somewhat higher false positive rate than the other methods. We hypothesise that this is because the larger standoff distances mean that more interest points occur in each image, raising the risk of false positive SIFT returns.

7.2 Number of views

The average number of views obtained before search terminated is presented in Figure 11(a); Figure 11(b) shows the same for successful outcomes only.

Indirect search, despite potentially "wasting" views looking for other objects, shows a marked advantage in view count, especially if failed searches are taken into account. Again, this is a result of the greater size of the view cones (due to larger objects) used in indirect search.

It is worth adding that indirect search was the only strategy to ever terminate because the posterior probability was too low, rather than because the maximum number of views was achieved. In these cases, the robot failed to detect the book even with a precise estimate of its location, and was in effect forced to conclude that it was not there at all. The uninformed and direct strategies in contrast were never able to cover sufficiently the prior distribution before the view limit was reached. Far from reaching the limit of $\mathbf{p}(\alpha_0)$ = 70%, after 15 views, D-search never got farther than just over 40% (from a starting value of 30%), and U-search barely even reached 35%. The problem is illustrated in Figure 7.2: After 15 views, because of their short range when searching uninformed for the small book, the robot's views have hardly covered any amount of space at all. In contrast, the large view cones used in the first stage of indirect search, as well as the concentrated posterior PDFs of its subsequent stages, mean that $\mathbf{p}(\alpha_0)$ will increase that much more quickly.



(a) All outcomes (b) Successful detection

Figure 11: Number of views obtained per run

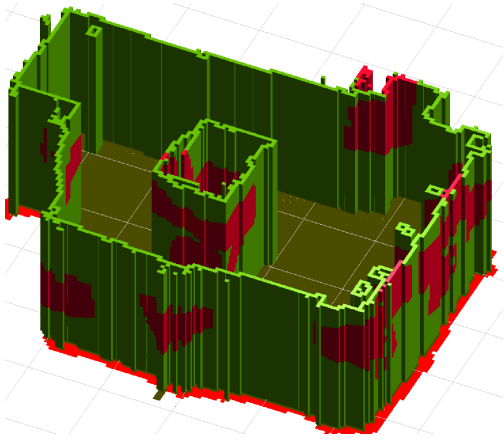


Figure 12: PDF change after 15 views of uninformed search. Red represents space where the robot had looked but not found the target object, hence the probability has decreased.

8 Summary and Discussion

In this paper we have proposed two idealized cognitive models for the core concept underlying English' "on", *viz.* mechanical support, and "in", *viz.* containment, in order to give us a functionally grounded models of the corresponding topological spatial relations. We have provided perceptual models which allows a robot to analyze a scene in terms of these relations using real sensory data. We have argued that spatial relations are a way to approach semantic perception and serve as a tool for qualitative reasoning and learning, top-down processes such as visual search, and linguistic interaction. We have also shown how we can make use of these spatial relations to realize indirect search in a principled way and verified via extensive experiments that it is superior to standard methods. We believe that these are important contributions to the field of semantic perception.

As the results presented in Sec. 7 indicate, spatial relations provide a powerful machinery in the context of semantic perception. However, needless to say, this work is just the beginning of what would be possible. As already discussed briefly in the introduction, we believe that spatial relations can help bridge the gap in the communication between humans and robots, providing a common ground for thinking about space in more qualitative terms. This is one of the paths we are currently pursuing. Also, in this work we have assumed that the robot was given the relations between objects. In a more complete system such information should be learned by the robot. We believe that spatial relations have an important role to fill in this, by means of providing an abstraction and thus making the learning processes more tractable, requiring less real training data and also opening up for acquiring such knowledge from humans directly or indirectly through databases as as the OMICS. A framework with spatial relations would be well suited for storing commonsense or typical knowledge, for example in the form of a Bayesian Network.

In this work we have made use of two topological spatial relations, namely "on" and "in". As mentioned before, there are many other relations that would be very useful, especially in the communication with humans. Results from e.g. psycholinguistics might help indicate which relations to choose, and how they might be modelled. Also, our implementation has been limited to box and plane shapes. While it can be readily extended to any convex 3D polyhedra, non-convex shapes require that further assumptions be made. Another possible future work is to define the active visual search problem as a planning problem in a similar fashion as in [23]. Planning on the geometric level of active visual search would result in an intractable amount planning states; one use for the present work might be to combine high level planning – which has the potential of bringing theoretical soundness to the problem – with low-level details such as occlusion and the geometry of the environment in order to find an optimal search plan.

The results also clearly show how much the efficiency of active visual search can be improved by moving away from brute force uninformed search and make use of the paradigm of indirect search building upon spatial relations. This is the main focus in this paper. One direction for future work would be to make use of more sensor data such as depth information from stereo cameras or shortly the new generation of depth

cameras to be able to lift the assumption that objects are located where the laser detects obstacles at its height.

Exploring the break-even point where direct informed search becomes more efficient than indirect search, along with automatically evaluating the optimal strategy to choose, are also possible future directions for research. Another interesting avenue for future investigation is to include the exploration of the rough geometry of space into the object search process. This would mean that the robot, in addition to selecting views for the camera, can select actions corresponding to exploring as yet unknown parts of space where no detailed priors can be defined. Currently, the object search is just a consumer of information from the spatial relations machinery. In the future these two subsystems should be connected bi-directionally so that information can feedback.

Some more general observations can also be made: The implementation is far from optimized and the performance can be greatly improved by, for example, parallelizing the computations and making use of a multi-core architecture. Most of the computations readily lend themselves to this. While the experiments in the paper clearly show the usefulness and power of our method and thus are sufficient for this paper, more experiments are needed to fully characterize the performance in different environments, with different objects etc.

The novel perceptual model implemented for "on" and "in" in this work assumes knowledge of the involved objects' geometry, poses and centers of mass. Whereas a human is able to estimate these quantites, even for novel objects, and/or extrapolate them based on experience, a robot may not always have access to good estimates from its visual system. Vision is not the focus of this work, however, and the soft nature of the applicability functions gives some robustness to poor visual information.

Acknowledgements

This work was supported by the Swedish Foundation for Strategic Research (SSF) through its Centre for Autonomous Systems (CAS), the EU integrated project ICT-215181-CogX, and the Swedish Research Council contract 621-2006-4520 (K. Sjöö). The support is gratefully acknowledged.

We would also like to thank Staffan Gimåker and Anders Boberg for providing the robotics community with the visualization tool peekabot
(http://www.peekabot.org) which we used extensively in our development and experiments. Special thanks goes to Staffan for his support. Finally, we acknowledge the support of CVAP staff during our experiments.

References

[1] "Openmind indoor commonsense," http://openmind.hri-us.com/, 2010.

[2] S. Vasudevan, S. Gächter, and R. Siegwart, "Cognitive spatial representations for mobile robots perspectives from a user study," In Proc. ICRA Workshop: Semantic Information in Robotics (ICRA - SIR 2007), Rome, Italy, 2007.

[3] R. Bajcsy, "Active perception vs. passive perception," in *Proc. 3rd Workshop on Computer Vision: Representation and Control*. Washington, DC.: IEEE Press, October 1985, pp. 55–59.

[4] T. D. Garvey, "Perceptual strategies for purposive vision," Artificial Intelligence SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Tech. Rep. 117, Sept. 1976.

[5] A. Herskovits, *Language and Spatial Cognition*. Cambridge University Press, 1986.

[6] K. Coventry and S. Garrod, *Saying, seeing and acting : the psychological semantics of spatial prepositions*. Hove, 2003.

[7] S. Levinson, "Language and space," *Annual Review of Anthropology*, 1996.

[8] J. O'Keefe, *The Spatial Prepositions*. The MIT Press, 1999, ch. 7.

[9] T. Regier and L. A. Carlson, "Grounding spatial language in perception: An empirical and computational investigation," *Journal of Experimental Psychology*, vol. 130, no. 2, pp. 273–2098, 2001.

[10] K. Lockwood, K. Forbus, D. Halstead, and J. Usher, "Automatic categorization of spatial prepositions," in *Proceedings of the 28 th Annual Conference of the Cognitive Science Society.*, 2006.

[11] J. Kelleher, "A perceptually based computational framework for the interpretation of spatial language," Ph.D. dissertation, Dublin City University, 2003.

[12] A. Cohn and S. Hazarika, "Qualitative spatial representation and reasoning: An overview," *Fundamenta Informaticae*, 2001.

[13] G. Lakoff, *Women, fire and dangerous things: what categories reveal about the mind*. University of Chicago Press, 1987.

[14] J. Piaget and B. Inhelder, *The Child's Conception of Space*. Routledge & Keagan Paul Ltd., 1956.

[15] S. Levinson and S. Meira, "natural concepts in the spatial topological domainadpositional meanings in crosslinguistic perspective: An exercise in semantic typology," *Language*, vol. 79, no. 3, 2003.

[16] L. Talmy, "Force dynamics in language and cognition," *Cognitive Science*, 1988.

[17] A. Torralba, M. S. Castelhano, A. Oliva, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search," *Psychological Review*, vol. 113, p. 2006, 2006.

[18] K. Welke, T. Asfour, and R. Dillmann, "Active multi-view object search on a humanoid head," in *ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 2041–2047.

[19] J. Winkeler, B. Manjunath, and S. Chandrasekaran, "Subset selection for active object recognition," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, p. 2511, 1999.

[20] M. Björkman and J.-O. Eklundh, "Vision in the real world: Finding, attending and recognizing objects," *International Journal of Imaging Systems and Technology*, vol. 16, pp. 189–208, 2006.

[21] T. Kawanishi, H. Murase, and S. Takagi, "Quick 3d object detection and localization by dynamic active search with multiple active cameras," *Pattern Recognition, International Conference on*, vol. 2, p. 20605, 2002.

[22] F. Saidi, O. Stasse, and K. Yokoi, "Active visual search by a humanoid robot," *Recent Progress in Robotics: Viable Robotic Service to Human*, vol. 16, pp. 171–184, 2009.

[23] G. Hollinger, D. Ferguson, S. Srinivasa, and S. Singh, "Combining search and action for mobile robots," in *ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 800–805.

[24] A. Andreopoulos and J. K. Tsotsos, "A theory of active object localization," in *IEEE International Conference on Computer Vision*, 2009.

[25] T. Deyle, H. Nguyen, M. Reynolds, and C. C. Kemp, "Rf vision: Rfid receive signal strength indicator (rssi) images for sensor fusion and mobile manipulation," in *IROS'09: Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 5553–5560.

[26] S. Ekvall, D. Kragic, and P. Jensfelt, "Object detection and mapping for service robot tasks," *Robotica: International Journal of Information, Education and Research in Robotics and Artificial Intelligence*, 2007.

[27] K. Sjö, D. Gálvez López, C. Paul, P. Jensfelt, and D. Kragic, "Object search and localization for an indoor mobile robot," *Journal of Computing and Information Technology*, vol. 17, no. 1, pp. 67–80, March 2008.

[28] J. K. Tsotsos and K. Shubina, "Attention and visual search : Active robotic vision systems that search," in *International Conference on Computer Vision Systems ICVS'07*. Washington, DC, USA: IEEE Computer Society, 2007, p. 539.

[29] L. E. Wixson and D. H. Ballard, "Using intermediate objects to improve the efficiency of visual search," *Int. J. Comput. Vision*, vol. 12, no. 2-3, pp. 209–230, 1994.

[30] Y. Ye, "Sensor planning for object search," Ph.D. dissertation, 1998.

[31] J. K. Tsotsos, "On the relative complexity of active vs. passive visual search," *International Journal of Computer Vision*, vol. 7, no. 2, pp. 127–141, 1992.

[32] M. Bar and S. Ullman, "Spatial context in recognition," *Perception*, vol. 25, pp. 324–352, 1993.

[33] M. Bar, "Visual objects in context," *Nature Reviews: Neuroscience*, vol. 5, pp. 617–629, August 2004.

[34] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communication on the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[35] W. Morris, I. Dryanovski, and J. Xiao, "3d indoor mapping for micro-uavs using hybrid range finders and multi-volume occupancy grids," in *RSS 2010 workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Zaragoza, Spain, July 2010.

[36] K. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems," in *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, 2010.

[37] D. Meger, P. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. Little, and D. Lowe, "Curious george: An attentive semantic robot," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 503–511, 2008.

[38] K. Sjöö, D. G. López, C. Paul, P. Jensfelt, and D. Kragic, "Object search and localization for an indoor mobile robot," *Journal of Computing and Information Technology*, vol. 17, no. 1, pp. 67–80, 2009, doi:10.2498/cit.1001182.

[39] D. Gálvez López, K. Sjö, C. Paul, and P. Jensfelt, "Hybrid laser and vision based object search and localization," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'08)*, 2008.

[40] A. Richtsfeld, T. Mörwald, M. Zillich, and M. Vincze, "Taking in shape: Detection and tracking of basic 3d shapes in a robotics context," in *Computer Vision Winder Workshop*, 2010, pp. 91–98.

[41] J. Folkesson, P. Jensfelt, and H. Christensen, "The m-space feature representation for slam," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1024–1035, Oct. 2007.