Exploiting Probabilistic Knowledge under Uncertain Sensing for Efficient Robot Behaviour

Author

Hanheide, M., Gretton, C., Dearden, R., Hawes, N., Wyatt, J., Pronobis, A., Aydemir, A., G¨obelbecker, M., Zender, H.

Published

2011

Conference Title

Twenty-Second International Joint Conference on Artificial Intelligence Proceedings

DOI

10.5591/978-1-57735-516-8/IJCAI11-407

Rights statement

Downloaded from

Link to published version

Griffith Research Online

Exploiting Probabilistic Knowledge under Uncertain Sensing for Efficient Robot Behaviour

M. Hanheide, C. Gretton, R. Dearden, N. Hawes, and J. Wyatt
University of Birmingham, England

A. Pronobis and A. Aydemir
KTH Stockholm, Sweden

M. Göbelbecker
University of Freiburg, Germany

H. Zender
DFKI Saarbrücken, Germany

Abstract

Robots must perform tasks efficiently and reliably while acting under uncertainty. One way to achieve *efficiency* is to give the robot common-sense knowledge about the structure of the world. *Reliable* robot behaviour can be achieved by modelling the uncertainty in the world probabilistically. We present a robot system that combines these two approaches and demonstrate the improvements in efficiency and reliability that result. Our first contribution is a probabilistic relational model integrating common-sense knowledge about the world in general, with observations of a particular environment. Our second contribution is a continual planning system which is able to plan in the large problems posed by that model, by automatically switching between decision-theoretic and classical procedures. We evaluate our system on object search tasks in two different real-world indoor environments. By reasoning about the trade-offs between possible courses of action with different informational effects, and exploiting the cues and general structures of those environments, our robot is able to consistently demonstrate efficient and reliable goal-directed behaviour.

1 Introduction

One dream of the AI community is to build a robot capable of performing tasks on demand in dynamic real-world environments like homes and offices. Such a robot must perform task and observation planning under uncertainty in pursuit of its current goals. It must do this while exploiting knowledge about the nature of the environments in which it is expected to operate. Towards realising the stated dream, this paper presents a robot system that uses a new planning approach to reason with new representations of space. Our approach integrates probabilistic models of common-sense conceptual knowledge, with models of the visual appearance of objects and of room categories, to represent an object search task. In order to allow the robot to effectively exploit this knowledge, we have developed a novel system for continual planning that automatically switches between using decision-theoretic and classical procedures to synthesise efficient action strategies.

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement No. 215181, CogX.

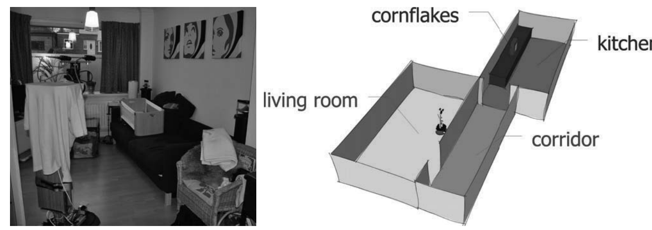


Figure 1: Our object search robot in a home environment composed of rooms of different categories. An example use case of the system is to find cornflakes located in the kitchen. The mobile robot is equipped with a laser scanner and a stereo camera rig.

We have implemented our approach on the mobile robot depicted in Fig. 1, and evaluated that system by having it perform *object search* tasks in real-world home and office environments. The objects it is able to search for are all *instances* of *categories*, e.g. a specific box of cornflakes in the kitchen, as depicted in Fig. 1, is an instance of the category of cornflakes boxes, which is itself a sub-category of cereal boxes. The robot uses structured representations of knowledge at this *conceptual* level – e.g. cereal boxes are often located in kitchens or dining rooms, and sofas are often located in living rooms. Such *relational structure* expresses generalisations across multiple environments, and can be naturally represented probabilistically in order to support intelligent decision making across multiple environments. We have compiled a common-sense knowledge base in an offline manner. Our two key novel contributions are:

1. A probabilistic conceptual map that combines general purpose and contingent spatial knowledge in a single structure, together with processes for creating, maintaining, and reasoning with it. This relational structure models the uncertain contingent knowledge the robot has about instances (e.g. what category of room it thinks room 1 is) in conjunction with its – also uncertain – common-sense conceptual knowledge (e.g. which types of objects are located in a particular category of room).

2. A switching continual planner that synthesises action strategies for the very large partially observable decision processes posed by the tasks we consider. Our approach is to switch between decision-theoretic and classical modes of planning at different levels of abstraction. The classical system quickly solves a determinisation of the problem at hand, interpreting probabilistic information in terms of a cost model. The decision-theoretic system quickly solves abstract decision problems derived using the classical plan and the probabilistic belief-state. Overall, this approach allows the system to exploit our rich representation of spatial knowledge, and generate intelligent behaviour under uncertainty in a timely manner.

2 Related Work

Probabilistic representations are employed for many localised functions in robots operating in the real world. For example, [Thrun *et al.*, 2000] use such representations in most of their system's individual components, but their robot behaviour is generated using a reactive controller rather than a domain-independent planner as here. [Kraft *et al.*, 2008] treat sensing deterministically and beliefs qualitatively during planning. We are not aware of any robot system that features both a unifying probabilistic representation, and a domain-independent planner which is able to reason quickly over that unified decision-theoretic model to generate behaviour.

Object search with mobile robots has been studied for almost 20 years, yet no previous system plans with probabilistic conceptual knowledge about both room and object categories. Instead, most dedicated systems treat the problem as a geometric one. For example, recently [Shubina and Tsotsos, 2010] propose how a robot can optimally locate an object in a mostly unknown 3D space. Closest to our approach is the work by [Aydemir *et al.*, 2011] who used probabilistic spatial relations and static properties of rooms to pose the object search problem as a fully-observable Markov decision process (MDP). This work employed background knowledge to inform an MDP planner of good locations (e.g. room1) to search for a particular object. Earlier work by [Galindo *et al.*, 2005] proposed to make this relationship bi-directional: objects give evidence for room categories, and room categories provide information about where objects can be found. In [Bouguerra *et al.*, 2007] this approach was extended to treat some of the conceptual knowledge as uncertain, although sensing here is restricted to object occurrence and the planner does not use a stochastic model of sensing. [Vasudevan and Siegwart, 2008] went beyond this to perform room categorisation using Bayesian reasoning about the presence of objects, but did not (as none of these did) include observation models for planning.

Compared to these existing approaches, we utilise a richer spatial representation combining visual room appearance, room geometry, presence of objects and the topological structure of space, extending our previous work [Pronobis *et al.*, 2010a] which only combined visual appearance of rooms and their geometry for the purpose of room categorisation. Also, our system is a successor to a robot that was able to exploit only deterministic conceptual and instance knowledge [Hawes *et al.*, 2011].

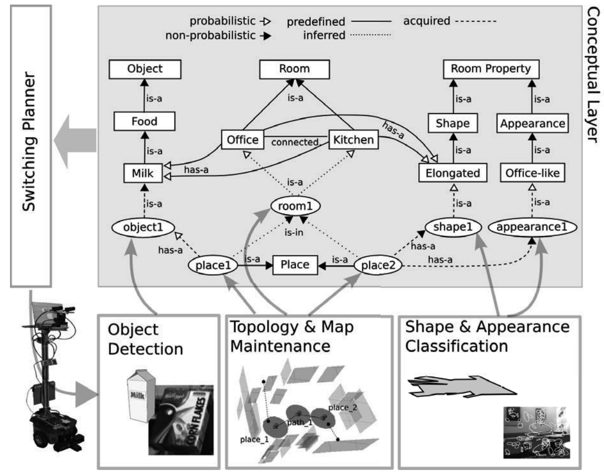


Figure 2: An abstract view of the processes and representations of the system. Sensing processes (at the bottom) discretise and categorise sensor input into instances (shown as ellipses) and acquired relations in conceptual layer. This layer also comprises knowledge about concepts (rectangles) of which only an excerpt in shown. The switching planner (cf. Sec. 4) reasons upon the state distribution given by the conceptual map.

3 Conceptual Map

The *conceptual map* realises the highest layer of the our qualitative spatial framework [Pronobis *et al.*, 2010b; 2010a]. This framework comprises several layers of abstraction from sensor readings, up to a topological map represented as a graph of interconnected *places* which each form a part of a *room*. On top of the topological map the framework includes the conceptual map populated by instances, of pre-defined concepts, generated by dedicated processes. An excerpt of a conceptual map for our object search task is shown in Fig. 2. The conceptual map is *relational*, describing common-sense knowledge as relations between concepts, and describing instance knowledge as relations between either instances and concepts, or instances and other instances. Relations in the conceptual map are either predefined, acquired, or inferred, and can either be deterministic or probabilistic. A non-existing relation in the conceptual map is thought of as having probability 0. An acquired relation is one that is grounded in observation and generated as a result of a sensing process. Predefined relations are given (and quantified in the case they are probabilistic) as part of a fixed ontology of default knowledge. Any inferred relations are the result of inference processes operating solely on the conceptual map.

In our implementation of the conceptual map, the concepts, and relations between these, were selected to enable our robot to reason about conceptual knowledge for efficient object search. The representation defines a taxonomy of concepts using hyponym relationships (is-a) as well as directed relations between rooms and objects (has-a). However, we also represent undirected associative relations (such as the connectivity between rooms) in our model. The processes that populate the model with instances and acquired relations between those are shown at the bottom of Fig. 2 and are as follows.

3.1 Sensing & Acting

In our system sensing is managed by a collection of processes which abstract from odometry data, laser scans and video sequences to maintain *instances* and the *probabilistic relations* which link these instances to concepts and other *instances*. We distinguish *continuous* and *active* sensing. The former is passive, continuously revising the robot's subjective beliefs about the world. It is lightweight, and does not require a planner that might schedule information gathering actions. In contrast, active sensing is deliberately planned for.

Mapping and Topology Maintenance is a continuous process that uses a SLAM algorithm [Folkesson *et al.*, 2007] to maintain metric and topological maps of the environment and localise the robot in those maps. It discretises space into metrically localised *places* approximately $1m$ apart (represented by discs in Fig. 5). It also maintains a navigation graph that supports movement from one place to another. Place existence and connectivity is treated deterministically in our current system. In order that topological places be interpreted with respect to higher level spatial concepts, mapping also features door frame detection from laser data. Using the non-monotonic reasoning approach of [Hawes *et al.*, 2011] places are grouped into rooms based on these detected door frames. The results of this continuously running process are instances of places and rooms with acquired connectivity relations.

Shape and Appearance Classification is achieved by continuous sensing of shape and appearance properties at topological places. Following [Pronobis *et al.*, 2010a], for a small discrete set of views at each place the robot senses low level features: (a) about the geometric shape from that view according to laser scans, and (b) about its visual appearance according to Composed Receptive Field Histograms. Those features are evaluated on the basis of Support Vector Machine (SVM) models of *specialised concepts* of "Room Property" – e.g. elongated, office-like, etc. Accumulated confidences gained from the SVM models across views are normalised to gain probabilities. These are represented in the probabilistic "is-a" relation that ties property instances to the specific concepts (cf. Fig. 2).

Object Detection is the only active sensing process in our system. It is triggered when the robot executes a visual sensing action. The underlying vision algorithms for object detection are from the BLORT toolkit [Mörwald *et al.*, 2010], and are applied to images from the robot's cameras. Object detection exhibits false positive and false negative detection rates that characterise the observation model for planning. Observation models allow the robot to reason that it might not have sensed an object despite it being perceivable, and vice-versa. The robot can then quantify the effects of active sensing processes on the conceptual map. For example, a detected object leads to the creation of a "has-object" relation for the specific instance the robot was looking for (cf. Fig. 2).

Actions in our system are all triggered by the planner. The planner typically solves two sub-problems: *navigation* and *local active visual search*. Navigation in the world is planned using the navigation graph defined by the connectivity relations. Movement between places is executed by the navigation component and includes local object avoidance. Local active visual search first requires an action to trigger the generation of discrete *viewpoints*. Following [Aydemir *et al.*, 2011], the generation action is executed as a Monte-Carlo sampling of local metric maps yielding information about the probability of object presence. Viewpoints are assigned an observation probability for a set of objects. The planner then reasons using actions to move to a viewpoint and trigger goal-directed object detection for appropriate objects.

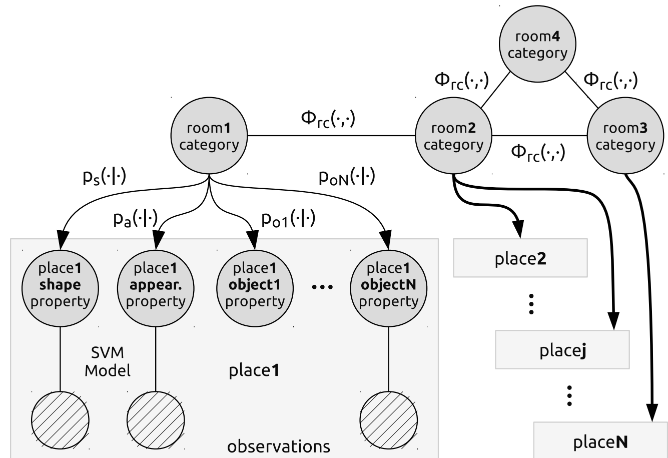


Figure 3: Structure of the chain graph model compiled from the conceptual map. The vertices represent random variables. The edges represent the directed and undirected probabilistic relationships between the random variables. The textured vertices indicate observations that correspond to sensed evidence.

3.2 The Chain Graph Representation

The conceptual map features probabilistic relations whose probability values cannot directly be acquired through sensing processes but have to be inferred (cf. Fig. 2). In order to support Bayesian inference in the conceptual map, the relational representation is compiled into a *chain graph* [Lauritzen and Richardson, 2002] representation, whose structure is adapted online according to the state of underlying topological map. Chain graphs provide a natural generalisation of directed (Bayesian Networks) and undirected (Markov Random Fields) graphical models, allowing us to model both "directed" causal (such as "is-a" relations) as well as "undirected" symmetric or associative relations (such as connectivity). The use of a chain graph allows us to model circular dependencies originating from possible loops in the topological graph, as well as direct use of the probabilistic relations between the concepts. In our implementation, chain graph inference is event-driven. For example, if an appearance property, or object detection alters the probability of a relation, inference proceeds to propagate the consequences throughout the graph. In our work, the underlying inference is approximate, and uses the fast Loopy Belief Propagation [Mooij, 2010] procedure.

An exemplary chain graph corresponding to the conceptual map shown in Fig. 2 is presented in Fig. 3. Each discrete place instance is represented by a set of random variables, one for each class of relation linked to that place. These are each connected to a random variable over the categories of rooms, representing the "is-a" relation between rooms and their categories in Fig. 2. Moreover, the room category variables are connected by undirected links to one another according to the topological map. Here, the potential functions $\phi_{rc}(\cdot,\cdot)$ describe the type knowledge about the connectivity of rooms of certain categories (e.g. that kitchens are more likely to be connected to corridors than to other kitchens).

The remaining variables represent: shape and appearance properties of space as observed from each place, and the presence of objects. These are connected to observations of features extracted directly from the sensory input. As explained in Sec. 3.1, these links are quantified by the categorical models of sensory information. Finally, the distributions $p_s(\cdot|\cdot)$, $p_a(\cdot|\cdot)$, $p_{o_i}(\cdot|\cdot)$ represent the common sense knowledge about shape, appearance, and object co-occurrence, respectively. They allow for inference about other properties and room categories e.g. that the room is likely to be a kitchen, because you are likely to have observed cornflakes in it.

3.3 Quantifying Probabilistic Relations

Our robot appeals to common-sense conceptual knowledge in order to act intelligently in indoor environments. That knowledge encapsulates, for instance, how likely it is that cereals will be found in kitchens, how hallways are usually long and offices visually cluttered, and how rooms of different types are typically connected. A question that remains is how we can quantify the probabilistic relations, such as "has-object", in Fig. 2. Our approach is to leverage common-sense knowledge available through the world wide web (WWW) to yield *object-location cooccurrence priors*.

In our system, object and location concepts are taken from the 'locations' database provided by the *Open Mind Indoor Common Sense* (OMICS)[1] project. Compiled with the express aim of making indoor mobile robots more intelligent, that database comprises 5,800 user-given associations between common everyday objects (ca. 2,900 unique categories) and their typical locations (ca. 500 unique categories). It does not, however, quantify the likelihood of object-location pairs. Where o is a specialisation of object (e.g. "milk") and l a location (e.g. "office"), we obtain *cooccurrence frequency estimates* by counting the number of *hits* an image search engine[2] returns when resolving "o in the l" queries for each of the 1.5 million object-location pairs from OMICS. Writing $\#q(o\&l)$ for the number of hits returned by that query, and $\#q(l)$ for the number when we query the noun term l, then the *cooccurrence prior* $c(o, l)$, that o is located in l, is given by Eq. 1.

$$c(o,l) = \left(\frac{\sqrt{\#q(o\&l)}}{\sqrt{\#q(l)}} \right)^B \tag{1}$$

In Eq. 1, $B{=}\frac{1}{2}$ for pairs that are said to occur together in the OMICS *locations* database, and is otherwise 1. We avoid using the raw frequencies from the search engine results to mitigate the problems of: (1) occluded objects being under-represented in image search queries – e.g., cups are stored in cupboards, and (2) image search queries are often biased to human interest, and omit the mundane and ordinary – e.g., ducks and baths are common, however faucets and baths are rarely mentioned together. We mitigate those problems by first applying the square root function to the counts. The B term has the effect of biasing the estimates to coocurrences that are deterministically represented in the OMICS database.

[1] `openmind.hri-us.com`, Honda Research Institute USA

[2] `images.bing.com`

4 Switching Continual Planner

To generate flexible goal-oriented behaviour our system employs a domain-independent planner. This takes a starting belief-state description compiled from the probabilistic conceptual map. From a planning perspective our mobile robot domain poses important but contrary challenges. On the one hand, planning and execution monitoring must be lightweight, robust, timely, and should span the lifetime of the robot. Those processes must seamlessly accommodate exogenous events, changing objectives, and the underlying *unpredictability* of the environment. On the other hand, in order to act intelligently the robot must perform computationally expensive reasoning about *contingencies*, and possible revisions of its subjective belief according to quantitatively modelled uncertainty in acting and sensing. Addressing specifically this second challenge, [Talamadupula *et al.*, 2010] identify *continual planning* in the presence of detailed probabilistic models as an important direction for research.

There has been much recent work scaling POMDP solution procedures to medium-sized instances. In the case of general domain-independent factored systems, the state-of-the-art scales to relatively small problems with 2^{22} states [Shani *et al.*, 2008].[3] At their limit, these procedures take over an hour to converge. For classes of POMDP that feature exploitable structures, for example, no actions with negative effects, problems with as many as 10^{30} states can be targeted by offline procedures [Brunskill and Russell, 2010]. Moving someway towards addressing all the challenges we have outlined, recent online POMDP solution procedures have been developed which can exploit highly approximate value functions – typically computed using a point-based procedure – and heuristics in forward search [Ross *et al.*, 2008]. These approaches are applicable in relatively small problems, and can require expensive *problem-specific* offline processing in order to yield good behaviours. A *very* recent and promising online approach for large POMDPs employs Monte-Carlo sampling to break the curse of dimensionality in situations where goal reachability is easy [Silver and Veness, 2010]. Although we believe it an interesting item for future work to pursue that direction, it should be noted that ease of goal reachability is not guaranteed in the problems we face.

Our work takes a concrete step towards addressing all the challenges we outlined. We have developed a *switching* domain-independent planning system that operates according to the continual planning paradigm [Brenner and Nebel, 2009]. It uses first-order declarative problem and domain representations, expressed in a novel extension of PPDDL [Younes *et al.*, 2005] called *Decision-Theoretic (DT)PDDL*, for modelling stochastic decision problems that

[3] Considering only room categories and distribution of objects, problems we consider in this paper have over $\sim 10^{27}$ states. The details of view points from local active visual search, and those of robot location further increase that figure. Therefore, not only because they are offline, but also because they have limited scalability these approaches are infeasible in our setting.

feature partial observability. In this paper we restrict our attention to DTPDDL models that correspond to deterministic-action goal-oriented POMDPs where all actions have non-zero cost[4] – i.e., an optimal policy can be formatted as a finite horizon contingent plan. Also, without a loss of generality we assume goals (i.e., conditions for reward) and action preconditions are conjuncts over *positive* propositions(/facts).

Our continual planning system *switches*, in the sense that the underlying planning procedure changes depending on our robot's subjective degrees of belief, and progress in plan execution. The system is continual in the usual sense that, whatever the session, plans are adapted and rebuilt online in reaction to changes to the planning model – e.g. when objectives are modified, or when our robot's path is obstructed by a door being closed. When the underlying planner is a deterministic sequential planner, i.e., a *classical* planner, we say planning is in a *sequential* session, and otherwise it is in a *DT* session. By autonomously mixing these two types of sessions our robot is able to be robust and responsive to changes in its environment, and make appropriate decisions in the face of uncertainty.

During a sequential session a rewarding *trace* of a possible execution is computed, in our experiments using a modified version of *Temporal Fast Downward (TFD)* [Eyerich *et al.*, 2009]. Taking the form of a classical plan, the trace specifies a sequence of actions that achieves the objectives following a deterministic approximation of the problem at hand, i.e., a *determinisation* [Yoon *et al.*, 2007]. A trace is a sequence of elements that are either: (i) actions from the DTPDDL description of the world, or (ii) atomic *assumptions*, modelled as deterministic actions, made about the truth value of facts that can only be determined at runtime – e.g., that the cereal is located in the kitchen. During sequential sessions the planner trades action costs, goal rewards, and determinacy, finding a highly valuable plan $\pi = s_0, a_0, s_1, a_1, .., s_N$ according to Equation 2.

$$V(\pi) = \prod_{i=1..N-1} \rho_i \sum_{i=1..N-1} \mathsf{R}(s_i, a_i) \qquad (2)$$

Here, ρ_i is the probability that the outcome, state s_{i+1}, of the i^{th} sequenced action a_i occurs, and $\mathsf{R}(s_j, a_j)$ is the instantaneous reward received for executing action a_j in state s_j. The system always begins with a sequential session, and once TFD produces a trace, plan execution proceeds by applying actions in sequence from that trace until $\rho_i < .95$ for the next scheduled action a_i. A DT session then begins which tailors sensory processing to determine whether the assumptions made in the trace hold, or which otherwise acts to achieve the overall objectives.

Because DT planning in large problems is too slow for our purposes, DT sessions plan in an abstract decision process determined by the current trace and underlying belief-state. The abstract process posed to the DT planner is constructed by first constraining as statically false all propositions except

[4]In the case of finite-horizon planning, POMDPs with stochastic actions can be compiled into equivalent deterministic-action POMDPs, where all the original action uncertainty is expressed in the starting-state distribution [Ng and Jordan, 2000].

(A) Partially constrained abstract belief-state

```
(:init (=(is-in Robot)kitchen)
 (.6(and(=(is-in cereal)kitchen)
       (.9(=(is-in milk) kitchen))
        .1(=(is-in milk)office))
  .4(and(=(is-in cereal)office)
       (.1(=(is-in milk)kitchen))
        .9(=(is-in milk)office)))
```

(B) Underlying DTPDDL belief

```
(:init (=(is-in Robot)office)
 (.6(and(=(is-in cereal)kitchen)
       (.9(=(is-in milk)kitchen))
        .1(=(is-in milk) office))
  .4(and(= (is-in cereal)office)
       (.1(=(is-in milk)kitchen))
        .9(=(is-in milk)office)))
 (.6(=(is-in cup)office)
  .4(=(is-in cup)kitchen)))
```

(C) Fully constrained abstract belief-state

```
(:init (=(is-in Robot)kitchen)
 (.6(=(is-in cereal)kitchen)))
```

Figure 4: Simplified examples of abstract belief-states from DT sessions.

those which are true with probability 1, or which are assumed true in the current trace. For example, if coffee cups are not necessarily in the kitchen, and the serial plan does not schedule actions whose outcomes are preconditioned on a cup being somewhere in particular (e.g., searching for or retrieving the cup from a view in the kitchen), then on first construction the states of the abstract process do not mention cups. Next, those static constraints are removed, one proposition at a time, until the number of states that can be true with non-zero probability in the initial belief of the abstract process reaches a given threshold (in our real-world experiments, 150 states). In detail, for each statically-false proposition we compute the *entropy* of the state assumptions of the current trace *conditional* on that proposition. Let X be a set of propositions and 2^X the powerset of X, then taking

$$\chi = \{ \bigwedge_{x \in X' \cap X} x \wedge \bigwedge_{x \in X \setminus X'} \neg x \mid X' \in 2^X \},$$

we have that χ is a set of conjunctions each of which corresponds to one truth assignment to elements in X. Where $p(\phi)$ gives the probability that a conjunction ϕ holds in the belief-state of the DTPDDL process, the entropy of X *conditional* on a proposition y, written $\mathrm{H}(X|y)$, is given by Equation 3.

$$\mathrm{H}(X|y) = \sum_{x \in \chi, y' \in \{y, \neg y\}} p(x \wedge y') \log_2 \frac{p(y')}{p(x \wedge y')} \qquad (3)$$

A low $\mathrm{H}(X|y)$ value suggests that knowing the truth value of y is useful for determining whether or not some assumptions X hold. When removing a static constraint on propositions during the abstract process construction, y_i is considered before y_j if $\mathrm{H}(X|y_i) < \mathrm{H}(X|y_j)$. For example, if the serial plan assumes the robot is in a kitchen, then propositions about the contents of kitchens, e.g. that there is a cup in the kitchen, are added to characterise the abstract process' states. If sensing scheduled during the DT session fails to find a cup in the room, then the kitchen assumption can be judged during DT deliberations. To the abstract model we add *disconfirm* and *confirm* actions that judge each assumption in the trace. These actions yield a small reward if the corresponding judgement is true and small penalty otherwise. Once a judgement action is scheduled for execution the DT session is terminated, and a new sequential session begins.

Abstract Process Example Following the syntax and semantics of PPDDL,[5] for a simplified object search task the current belief is described by the expression in Fig. 4B. This admits 8 states with non-zero probability. For example, with probability .324 the robot is in the office, cereal and milk are in the kitchen, and a cup is in the office. Suppose a serial session plans: (1) to relocate the robot to the kitchen, (2) observe the cereal, and (3) report that cereal is located in the kitchen to a user. Characterising the abstract problem posed to the DT session at step (2), Fig. 4C gives the belief in the case where all static constraints hold. Taking assumption X to be `(=(is-in cereal)kitchen)`, in relaxing static constraints the following entropies are calculated:

```
.47 = H(X|(=(is-in milk)office))
    = H(X|(=(is-in milk)kitchen))
.97 = H(X|(=(is-in cup)office))
    = H(X|(=(is-in cup)kitchen))
```

Therefore, the first static constraint to be relaxed is `(=(is-in milk)office)`, or equivalently `(=(is-in milk)kitchen)`, giving a refined abstract belief state depicted in Fig. 4A. Summarising, if the DT session is restricted to belief-states with fewer than 8 elements, then the starting belief-state of the DT session does not mention a "cup".

5 Experimental Evaluation

To evaluate the implemented representational and planning techniques, we first analysed our robot system performing an object search task in two different real-world environments: a larger office (O, 13 places in 3 rooms) and a smaller home (H, 7 places in 3 rooms). A sketch of the object search setting in environment H is depicted in Fig. 1. Our evaluation compares the full system described in this paper, exploiting probabilistic conceptual knowledge and evidence from shape and appearance classification, to a baseline system that cannot make use of the conceptual knowledge. We refer to these as the "full" and "lesioned" systems respectively. In the lesioned system continuous sensing of shape and appearance properties is disabled, emulating the limited reasoning available in our previous system [Hawes *et al.*, 2011]. Therefore it can neither use these properties to infer the categories of rooms nor can it exploit conceptual knowledge about object-location co-occurrence.

In all runs, a box of cornflakes (the object to search for) was placed in the environment among many other objects belonging to the nine categories the robot has been trained to detect. In the experiments, only a subset of all object-location co-occurrence frequencies consisting of 152 relations between the 19 selected object concepts (cornflakes among them) and seven given room concepts (among them kitchen, living room, corridor, and office) was employed. In a first set of runs directly comparing the full system (FC) to the lesioned case (LC), the box of cornflakes was in the kitchen, which is the canonical location for this type of object according to the common-sense conceptual knowledge

[5]Omitting the "probabilistic" string at the start of the corresponding PPDDL blocks to keep the descriptions small.

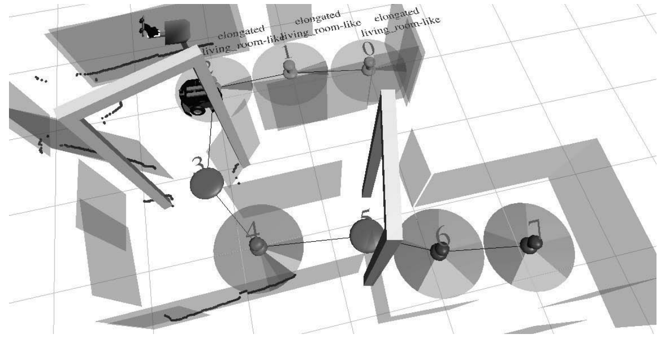


Figure 5: Environment H with numbered *places*, and pie charts indicating probabilities of room categories (yellow=living room, red=kitchen, green=corridor, blue=office, grey=others). The labels attached to place node in the living room (upper right) indicate the most likely values in the distribution of classified *shape* (top) and *visual* (bottom) properties, respectively. Detected doors, used for room partitioning, are shown as door frames. The kitchen is at the lower right.

conf.	obj. loc.	lesion	♯succ./♯tot. H	♯succ./♯tot. O	avg. time H	avg. time O
FC	kitchen	no	10/10	5/6	5.8min	6.8min
LC	kitchen	yes	9/10	5/5	11.3min	13.5min
FNC	liv. room, office	no	3/3	n/a	10.2min	n/a

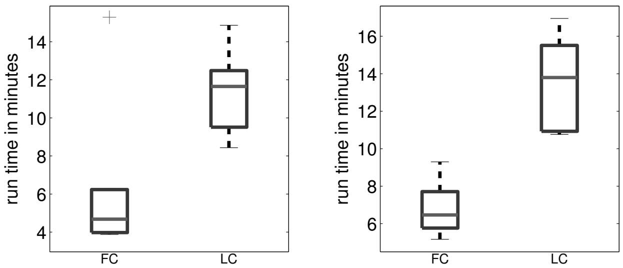
Table 1: Runtimes for the three cases tested: full system (FC) and lesioned system (LC), both with object in canonical position; NFC: full system with object in non-canonical position. Total time to solve the task reported in minutes. The FNC case was only tested in environment H.

(with a probability of $P(cornflakes|kitchen) = 0.336$). In a second set of runs, the object was at a non-canonical location ($P(cornflakes|living\_room) = 0.035$) to test the full configuration (results denoted as FNC).

Hypotheses The hypotheses leading to this study design are that (i) the exploitation of the conceptual knowledge in the full system will enable the robot to achieve the task quicker in canonical cases when compared to the lesioned system in the same experimental setup, (ii) although more efficient in the average case, the system will be robust in the presence of sensing errors, and (iii) that the system will still be able to achieve its goal, even relatively efficiently, in non-canonical setups. In all runs, before the robot was given the goal to find the object, it performed a short exploration of adjacent places to sense room properties in order to infer the category of the room (if this evidence was not lesioned). The acquired map after this short exploration for the H environment is shown in Fig. 5. The robot has already sensed room properties and hence evidence about the category of the room it is in is available in the conceptual map when making a first plan.

Quantitative Results from Real-World Experiments The cornflakes box was found by the robot in 32 of the 34 runs. In the two failed runs the robot maneuvered itself into a corner of the room and required human intervention. The total execution time of the successful FC and LC runs are plotted in Fig. 6.[6] Hypothesis (i) claims that the robot is able to exploit the evidence gained from perceiving its environment by integrating this with conceptual knowledge about the commonalities of such environments. That our system does this

[6]A video available at `http://youtu.be/0QcmSDZR-c4` illustrates an exemplary FC run in the H environment.



(a) env. H.
(b) env. O.

Figure 6: Box and whisker diagrams of total runtime to achieve the given task in two environments comparing the 'full (FC)' to the 'lesioned (LC)' case. In the FC case for environment H one run took longer than 14 minutes indicated by the outlier.

is confirmed by a significant difference (Mann-Whitney test $p < 0.01$ for both environments) in average runtime reported in Tab. 1 for these two configurations. Looking at the typical sequence of actions for the FC configuration it becomes apparent that planning inferred lower costs for driving into the kitchen to begin searching (despite that being an extra distance to travel without looking for objects). This observation also explains the relative improvement of FC in the larger O environment, comprising larger space that has to be searched exhaustively, being comparatively higher than in H.

In all our runs, total running time was dominated by action execution with planning being a distant second. The total time spent on planning was between 16.9 seconds for the FC runs in the H environment and 64.8 seconds for the LC configuration in the larger O environment. The time was divided roughly equally between the serial and DT sessions. The time spent on planning *per planner call* was slightly higher in the FC configurations, but this was offset by a much lower number of calls in the full configuration (on average 6 compared to 13 for H, 5.6 to 20.6 for O).

Qualitative Discussion of Results The results comparing FC and LC runs could admittedly have been obtained using a system that *determinsitically* exploits structural knowledge about cornflakes being found in kitchens instead of making use of the probabilistic formulation of the knowledge. Hence, the results so far only confirm hypothesis (i). With regard to hypothesis (iii) we can confirm that the robot was able to solve non-canonical configurations 100% of the time. In these runs, the robot also first searched in the kitchen before returning to the other room and eventually finding the object there. A system entirely dogmatic about the cornflakes being in kitchens (having modeled this relation as deterministic) would not have been able to consider this alternative. Another interesting case we encountered is evident in Fig. 6(a). The single outlier in this figure is related to hypothesis (ii), indicating that that our system can cope with uncertainty in sensing. In this case the robot also first drove to the kitchen (following its initial sequential plan), entered a DT session to find the object, but eventually failed to detect the object (due to an object detection false negative). Hence, the DT session disconfirmed the original assumption that was made. Accordingly a plan was created that drove the robot back to the living room to continue its search there; again due to the remaining probability of finding objects also in non-canonical locations. However, after a comprehensive search, the likelihood of finding the object in the kitchen by looking again became higher, so the robot went back and eventually found it. In general, realistically, object detection was very reliable in our system, with observation probabilities estimated as .05 for false positives and false negatives. Accordingly, we only observed one such case of a sensing error in our runs. In order to assess the potential of the switching planner in greater detail we conducted further experiments in a simulated setting, where we were in control of sensing characteristics.

Planning with Noisy Sensing We integrated our switching planner in an enhanced version of the MAPSIM simulation environment [Brenner and Nebel, 2009] and performed a detailed experimental evaluation, comparing switching using our DT procedure with a greedy dual-mode [Cassandra *et al.*, 1996] *baseline* we called "cp". For experimentation in simulation the base planner during sequential sessions is a cost-optimising version of *Fast Downward* [Helmert, 2006]. For the dual-mode baseline, when a scheduled action triggers a switch to a DT session, the system plans to a single entropy reduction action whose execution can provide evidence regarding the truth value of an assumption from the current trace. Control is returned to a new sequential session as soon as a sensing action is executed.

We evaluate those approaches in simulation on a number of tasks, including a 6-room environment comprising 26-places and 21-objects. In simulation we considered 3 levels of reliability in sensing: *reliable* sensors have a .1 probability of a false negative, *semi-reliable* have a chance of 0.3 of false negative and 0.1 of false positive, and *noisy* sensors with probabilities of 0.5 and 0.2 respectively. Our evaluation examines DT sessions with initial belief-states admitting between 20 (written "dt 20") and 100 abstract states with non-zero probability. We run 50 simulations in each configuration, and plot quality data in Fig. 7 for observed behaviours in the 6- room environment where the goal is to find and report the location of a target object to a user. Here, compared to the simple greedy sensing strategy of *cp*, we have that DT sessions yield a higher rate of success, and lower expected cost of achieving the goal. Moreover, as the reliability of sensing degrades there is a clear benefit in performing expensive DT planning. Although there is insufficient space to present the results here, we observed that the overall time spent planning increases linearly as we move from *cp* to progressively refined abstractions in DT sessions.

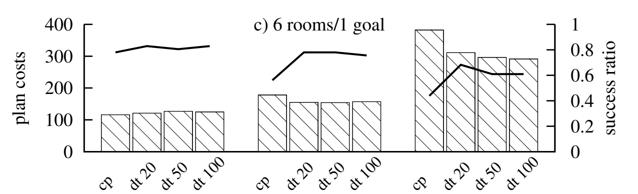


Figure 7: Average run cost (bars) and portion of successful runs (line) in simulation. For the left plot sensing is *reliable*, middle is *semi-reliable*, and right is *noisy*.

6 Conclusion

We presented a mobile robot system that integrates two original approaches for representing and reasoning about uncertainty. The first is a conceptual map, a representation of space that combines knowledge about its qualitative structure (e.g. a topological map), with probabilistic knowledge (e.g. that cereal boxes are found in kitchens 33% of the time). The second is a continual planning and execution monitoring system that employs *switching* to plan for *very* large partially observable problems that are posed by this representation. It is important to note that the *integration* of these approaches is crucial to the success of our work. Without the novel planner, the representation would not be capable of influencing behaviour. Without the novel representation, the planner would not be able to reason over both probabilistic instance *and* conceptual knowledge at the same time. We evaluated this combination in our robot in two real-world environments, and found that it is able to yield efficient and robust behaviours in an object search task.

References

[Aydemir *et al.*, 2011] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *Proc. Int. Conf. Robotics and Automation (ICRA)*, 2011.

[Bouguerra *et al.*, 2007] A. Bouguerra, L. Karlsson, and A. Saffiotti. Handling uncertainty in semantic-knowledge based execution monitoring. In *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*, pages 437–443, November 2007.

[Brenner and Nebel, 2009] M. Brenner and B. Nebel. Continual planning and acting in dynamic multiagent environments. *J. Autonomous Agents and Multiagent Systems*, 19(3):297–331, 2009.

[Brunskill and Russell, 2010] E. Brunskill and S. Russell. Rapid: A reachable anytime planner for imprecisely-sensed domains. In *Proc. Uncertainty in Artificial Intelligence (UAI)*. AUAI Press, 2010.

[Cassandra *et al.*, 1996] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*, pages 963–972, 1996.

[Eyerich *et al.*, 2009] P. Eyerich, R. Mattmüller, and G. Röger. Using the context-enhanced additive heuristic for temporal and numeric planning. In *Proc. Int. Conf. Automated Planning and Scheduling (ICAPS)*, 2009.

[Folkesson *et al.*, 2007] J. Folkesson, P. Jensfelt, and H. I. Christensen. The m-space feature representation for SLAM. *IEEE Trans. Robotics*, 23(5):1024–1035, October 2007.

[Galindo *et al.*, 2005] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernandez-Madrigal, and J. Gonzalez. Multi-hierarchical semantic maps for mobile robotics. In *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*, pages 2278–2283, August 2005.

[Hawes *et al.*, 2011] N. Hawes, M. Hanheide, J. Hargreaves, B. Page, H. Zender, and P. Jensfelt. Home alone: Autonomous extension and correction of spatial representations. In *Proc. Int. Conf. Robotics and Automation (ICRA)*, 2011.

[Helmert, 2006] M. Helmert. The fast downward planning system. *J. Artif. Intell. Research*, 26:191–246, 2006.

[Kraft *et al.*, 2008] D. Kraft, E. Başeski, M. Popović, A. M. Batog, A. Kjær-Nielsen, N. Krüger, R. Petrick, C. Geib, N. Pugeault, M. Steedman, T. Asfour, R. Dillmann, S. Kalkan, F. Wörgötter, B. Hommel, R. Detry, and J. Piater. Exploration and planning in a three-level cognitive architecture. In *Proc. Int. Conf. Cognitive Systems (CogSys)*, 2008.

[Lauritzen and Richardson, 2002] S. L. Lauritzen and T. S. Richardson. Chain graph models and their causal interpretations. *J. Roy. Statistical Society, Series B*, 64(3):321–348, 2002.

[Mooij, 2010] J. M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *J. Mach. Learn. Res.*, 11:2169–2173, August 2010.

[Mörwald *et al.*, 2010] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze. BLORT – The Blocks World Robotic Vision Toolbox. In *Proc. ICRA Workshop Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[Ng and Jordan, 2000] A. Y. Ng and M. I. Jordan. PEGASUS: A policy search method for large MDPs and POMDPs. In *Proc. Uncertainty in Artificial Intelligence (UAI)*, pages 406–415. Morgan Kaufmann Publishers Inc., 2000.

[Pronobis *et al.*, 2010a] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *Int. J. Robot. Res.*, 29(2-3):298–320, February 2010.

[Pronobis *et al.*, 2010b] A. Pronobis, K. Sjöö, A. Aydemir, A. N. Bishop, and P. Jensfelt. Representing spatial knowledge in mobile cognitive systems. In *11th International Conference on Intelligent Autonomous Systems (IAS-11)*, Ottawa, Canada, August 2010.

[Ross *et al.*, 2008] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa. Online planning algorithms for POMDPs. *J. Artif. Int. Res.*, 32:663–704, July 2008.

[Shani *et al.*, 2008] G. Shani, P. Poupart, R. Brafman, and S. E. Shimony. Efficient add operations for point-based algorithms. In *Proc. Int. Conf. Automated Planning and Scheduling (ICAPS)*, pages 330–337, 2008.

[Shubina and Tsotsos, 2010] K. Shubina and J. Tsotsos. Visual search for an object in a 3D environment using a mobile robot. *Computer Vision and Image Understanding*, 114(5):535–547, 2010.

[Silver and Veness, 2010] D. Silver and J. Veness. Monte-carlo planning in large POMDPs. In *Proc. Conf. Neural Information Processing Systems (NIPS)*, 2010.

[Talamadupula *et al.*, 2010] K. Talamadupula, J. Benton, S. Kambhampati, P. Schermerhorn, and M. Scheutz. Planning for human-robot teaming in open worlds. *ACM Trans. Intell. Syst. Technol.*, 1:14:1–14:24, December 2010.

[Thrun *et al.*, 2000] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000.

[Vasudevan and Siegwart, 2008] S. Vasudevan and R. Siegwart. Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robot. Auton. Syst.*, 56:522–537, June 2008.

[Yoon *et al.*, 2007] S.-W. Yoon, A. Fern, and R. Givan. FF-replan: A baseline for probabilistic planning. In *Proc. Int. Conf. Automated Planning and Scheduling (ICAPS)*, 2007.

[Younes *et al.*, 2005] H. L. S. Younes, M. L. Littman, D. Weissman, and J. Asmuth. The first probabilistic track of the international planning competition. *J. Artif. Intell. Res.*, 24:851–887, 2005.