J Intell Robot Syst (2018) 90:45–63
https://doi.org/10.1007/s10846-017-0643-1

A Bayesian based Methodology for Indirect Object Search

Patricio Loncomilla[1] · Javier Ruiz-del-Solar[1] · Marcelo Saavedra A.[1]

Received: 1 December 2016 / Accepted: 1 June 2017 / Published online: 7 October 2017

Abstract The main goal of this paper is to propose a Bayesian based methodology for implementing robot informed search for objects. The methodology uses convolutions between observation likelihoods of secondary objects and spatial relation masks for estimating the probability map of the object being searched for, and also a search procedure that uses this probability map. A method for computing complex spatial relation masks by using a basis composed of basic relation masks and a database of co-occurrences of objects is used. Each basic relation mask corresponds to a qualitative spatial relation (QSR), such as: 'very near', 'near', or 'far'. The search procedure takes into account the probability that the main object can be in different regions on the map and the distance to those regions. Also, the object search procedure is able to detect objects and generate new plans while moving. The proposed methodology is compared with uninformed and alternative informed search approaches using simulations and real-world experiments with a service robot. In simulations, the use of the proposed methodology increases the detection rate from 28% (direct uninformed search) to 79%, when the main object can be detected within a maximum distance of 1 meter. In the real world experiments, the use of the proposed methodology increases the detection rate from 40% (direct uninformed search) to 87% when using convolutions with soft masks, global search, and information on the positive detection of secondary objects. The detection rates obtained when using the proposed methodology are also much higher than those obtained by alternative informed search methods, both in the simulated and in the real-world experiments.

Keywords Semantic search · Informed search · Active search · Robot indirect search for objects

1 Introduction

The ability to search for objects in complex environments is fundamental for mobile service robots. A simple approach for performing this task consists of generating a set of consecutive viewpoints that maximize the probability of finding the object. This approach requires that the robot explores the whole environment in searching for the object. If the object to be detected is small, or if the environment is large, this exhaustive search approach will require a very large number of viewpoints for covering the environment.

However, a better searching approach can be applied if the fact is exploited that some sets of objects have specific spatial relations, mainly because they are all needed by humans for performing some defined tasks (e.g. cooking, or cleaning). These objects tend to appear close to each other as they have a particular semantic relation, making it possible to infer the presence of an object A, given an object B. For example, when looking inside a kitchen, it can be noted that the object "coffee cup" is often very near the object "coffee maker". Thus, when the main searched object has a known spatial relation with other objects in the environment based on its use, the detection of these

✉ Patricio Loncomilla
ploncomi@ing.uchile.cl

Javier Ruiz-del-Solar
jruizd@ing.uchile.cl

Marcelo Saavedra A.
hiperchelo@gmail.com

[1] Advanced Mining Technology Center, Department of Electrical Engineering, Universidad de Chile, Santiago, Chile

intermediate/secondary objects can be used to improve the search procedure for the primary/main one. Search methods that are able to use information about secondary objects for improving their performance are named informed object search methods. As a particular case, methods that first look for an intermediate object in order to find a main object are named indirect search methods. Informed search methods are able to overcome direct search when spatial relations between objects are known. Estimation of spatial relations requires additional work when compared to direct search methods. Also, errors in estimations or modeling of spatial relations can cause informed search to perform worse than direct search. Then, informed search methodologies must consider procedures for estimating the spatial relations using real-world data.

In this context the main goal of this paper is to propose a Bayesian based methodology for implementing indirect search for objects. The methodology uses convolutions between observation likelihoods of secondary objects and spatial relation masks for estimating the probability map of the object being searched for, and a search procedure that uses this probability map. The methodology is intended to be used by service robots operating in human environments where the semantic relationships between objects can be exploited.

This work makes two main contributions. The first contribution is the development of a family of Bayesian algorithms that enable integrating information about detections of secondary objects onto the probability map of the main object by using fast, focalized convolutions. They are based on the convolution of the observation likelihood of the secondary object over the map with a spatial relation mask. The result of the convolution is an observation likelihood for the main object, which is integrated onto the probability map. The masks can represent arbitrary spatial relations. The second contribution is a region-based global search procedure for selecting the next viewpoint of the robot, which takes into account the probability that the main object can be in different regions on the map and the distance to those regions. The planning procedure is fast to compute, enabling the robot to detect objects and reevaluate its navigation goal while moving.

The proposed methodology is compared with alternative indirect search approaches using simulations and real-world experiments with a service robot. Simulations are included because they allow performing replicable experiments and analyzing/simulating specific situations.

This work is an extension of that reported in [1, 2]. In this improved version, new search strategies are proposed; a more extensive analysis of the proposed methodology is carried out; experiments with a service robot operating in the real world are included; and a more extensive comparison with other indirect search methods is presented.

The work is organized as follows: In Section 2, the state-of-the-art related to the present work is described. In Section 3, the proposed methodology for indirect object search is explained. In Sections 4 and 5, a comparison of the proposed methodology with alternative indirect search approaches in simulations and in the real world, respectively, is carried out. Finally, in Section 6 conclusions of this work are drawn.

2 Literature Review

The search for objects in real environments is a complex task, which has been addressed using different approaches. In the case of informed search, in which the presence of secondary objects is considered for guiding the active search of the primary objects, a semantic modeling of the environment is also needed. In the following Sections, key papers related to the representation/modeling of the environment, and the planning aspects of the object search process are reviewed.

2.1 Representation of the Environment

Techniques used for representing the environment and the spatial relations among objects are highly diverse. However, most of the approaches use spatial relations, co-occurrences, and/or hierarchical structures.

- ***Spatial relations*** Objects in indoor environments, which are related to the same task, tend to appear following certain spatial patterns. As an example, keyboards tend to appear in front of monitors. Spatial relations are useful for predicting poses of searched objects when poses of secondary objects are known. In 1976, Garvey [3], recognized the usefulness of spatial relations for performing indirect search. Relations enable to constrain windows containing objects in the image domain using relations like “above” or “left-of”. However, spatial relations in the 3D space were not considered. In 2011, Kasper et al performed a study on spatial relations with three-dimensional images using a Kinect sensor [4]. They created a database using nine different office space settings with a total of 168 objects in 35 object classes. They then calculated the distances between different objects. Spatial relations are discretized both on distance and orientation space.

Methods that are able to use common-sense knowledge have been developed in the last years. This approach is derived from previous works of Randell (1992) [5] and Cohn (2001) [6], in which qualitative spatial relations (QSRs) are described as topological relations between regions representing objects. The approach was extended in

2014 by Cohn [7] for being able to include orientation information inside the qualitative spatial relation framework. QSRs have the potential to represent spatial relations by using human-like concepts. However, these works are theoretical and is not straightforward to implement them on a mobile robot.

In 2010, Aydemir et al. [8] developed a method for representing qualitative spatial relations in 3D by sampling positions of possible sought objects from detections of secondary objects. A model for the spatial relation "on" was handcrafted and enabled the robot to perform indirect object search. A SIFT object detector was used for detecting objects. The system was successful as objects are usually on horizontal surfaces, with the supporting object being larger than the supported one. In 2011, Aydemir et al. [9] improve the system, by using two spatial relations "in" and "on", and also improving the action selection process. In 2014, Kunze et al. [10] used QSRs between a landmark object and the object to be found. The relations being considered are directional (*left-of, right-of, in-front-of, behind-of*) and distance relations (*close-to* and *distant-to*). Each basic relation is represented by a Gaussian, and then any arbitrary relation can be decomposed into a sum of the basic relations and represented as a sum of Gaussians. The system assumes that the robot knows: a 2D map, a 3D map, the set of known landmarks, and the set of QSRs needed by the system to work. In 2014, Kunze [11] presented a system that is able to learn both QSRs from desktop scenes, and a set of useful landmarks for each object. A dataset of 47 scenes containing 437 object instances and 1,798 labeled spatial relations is used for learning the QSRs. Then, 500 desks are simulated by selecting sets of objects that are compatible with the estimated QSRs for data augmentation purposes. The set of useful landmarks for each object type is determined by analyzing the entropy of the distribution representing the relative positions of candidate landmarks an each object. In 2016, Li [12] proposed a novel system for learning and describing object relations between objects in an unsupervised fashion. An object is represented by an orientated bounding volume with six faces. The centroids of each face are used for describing the object. By computing differences between the positions of the centroids of two different objects, feature vectors describing spatial relations are computed. By applying clustering to the features, spatial relations can be learnt in an unsupervised way. However, this work is not yet applied to object search.

In summary, spatial relations are useful for inferring poses of sought objects from information related to other objects or locations.

- ***Co-occurrences*** A simple form for representing statistics about objects is to count the frequency of pairs of objects co-occurring in the same images. Also, statistics about co-occurrences of objects and places can be used. In 2009, Viswanathan et al. [13] proposed an approach for learning place models from objects. Co-occurrences between places and objects are computed by using marked images from the LabelMe database, designed by Russell et al. [14]. They train an automatic classifier of places, based on the presence of the detected objects, to infer the probability that the other objects are there, and the kind of place (e.g., kitchen or office) is seen in the setting. In 2012, Zhou et al. [15] proposed a system able to acquire commonsense knowledge about object locality (CSOL) by using both web mining (search engine querying) and a professional database for estimating co-occurrences between objects and locations. The method is tested in a real robotic platform. In 2013, Elfring et al. [16] proposed an active object search method that is based on co-occurrences of pairs of objects. The system is also able to infer chains of intermediate objects for searching objects inside a room. In 2015, Riazuelo [17] proposed a cloud platform named RoboEarth. Two functionalities provided by this system are the ability of creating semantic maps of the environment and of executing search tasks. Co-occurrences between objects and locations are used for creating object search plans. Thus, co-occurrences provide simple models that are useful for performing active search. However, they lack precise information about the spatial configuration of objects.
- ***Hierarchical map structures*** Co-occurrences and spatial relations provide useful information for informed object search. However, robot's environment need to been described as a metrical, topological and/or semantical representation for enabling the robot to navigate and then to search for objects. In 2005, Galindo et al. [18] performed a study based on 2D data, combining metric, topological, and semantic aspects on a map. In addition, they proposed a method for learning these semantic representations from sensory data. In 2007, Vasudevan et al. [19] attempted to create a spatial representation in terms of objects, by encoding typical household objects and doors within a hierarchical probabilistic framework. They used a SIFT [20] based object recognition system, and a door detection system based on lines extracted from range scans. They also proposed a conceptualization of different places, based on the objects that were observed inside them. In 2015, Hanheide et al. [21] proposed a system that is able to execute plans by organizing the knowledge of the robot about the task and their environment in three layers: an instance knowledge layer, a default (commonsense) layer, and a diagnostic knowledge layer. Each layer is able to modify lower layers. The default layer is used for generating plans, which are

stored in the instance knowledge layer. If the information is insufficient for deciding actions, *assumptions* are made by using information from the default layer. If the plan fails, the diagnostic knowledge layer modifies the assumptions in order to generate a new plan. Hierarchical map structures are useful for storing information about the environment with different levels of abstraction, ranging from occupancy maps and positions of objects up to high-level representations like common-sense knowledge about the environment. They are also useful for enabling the robot to generate plans involving sequences of actions needed for reaching a goal.

2.2 Planning for Object Search

Planning for object search is highly dependent on the representation used for the environment. Then, each work considered in the review has a different planning mechanism. However, it is possible to describe aspects involved in most of the approaches.

The search mechanisms can be classified by the kind of planning performed. They can be greedy (computing an optimal viewpoint at each time step) or consider several future time steps. In this work, greedy planning strategies are used as the base for performing object search.

In 1976, Garvey [3] proposed the idea of indirect search, i.e. searching for an intermediate object that maintains a particular spatial relation, based on their similar use, with the object being searched for. A greedy strategy was used for active vision, based on detecting secondary objects in the image space, and using them for selecting window hypotheses about the sought object. In 1994, Wixson et al. [22] analyzed theoretically indirect search in a 3D space, using also a greedy strategy. In 1999, Ye and Tsotsos [23] used a grid for representing a probability map about presence of the sought object. This system is based on direct search. In 2009, Viswanathan et al. [13] used a model composed of a metric map, detected objects, and object clusters. Also, co-occurrences between places and objects must be available. The system is able to generate paths for searching unobserved objects in a greedy fashion. In 2010, Shubina and Tsotsos [24] proposed an algorithm that considers the cost and effect of different actions with different types of prior knowledge. Map initialization can consider or not the known secondary objects. Planning is performed by selecting the optimal viewpoint at each time step. In 2010, Aydemir et al. [8] proposed a system that was able to search objects by using indirect search, and generating hypothesis about the sought object by sampling the spatial relation given a detected secondary object. The next viewpoint is selected by a greedy coverage approach, which is based on a probability map of the sought object, or in the point cloud of hypotheses when a secondary object is observed.

2.3 Object Search Algorithms

As mentioned, a wide variety of indirect object search algorithms have been proposed [3, 8, 9, 13, 22–25]. Systems based on QSRs have proved useful for conducting informed search. However, the use of these spatial relations is not straightforward because the probability map and the spatial relation are represented in different ways. Usually, the probability map is stored as a grid map, while spatial relations are represented as functions, by using sums of Gaussians, or sampling procedures. Then, for using spatial relations, the distribution needs to be sampled every time it needs to be used.

The methodology presented in this paper is an extension of the one proposed by Loncomilla et al. [1, 2]. That work is able to represent arbitrary spatial relations by using spatial relation masks. Also, each spatial relation mask can be decomposed as a combination of basic QSR masks. All of the information about detections of secondary objects is stored in a unique probability map, by computing fast, localized convolutions between the likelihood of the detections of secondary objects and the spatial relation mask. Also, in that work both object detection and planning are performed while the robot is moving, then it is not necessary to reach a goal before re-planning the next actions. Then, that work presents contributions that are complementary to the state of the art. In the present work, an improved methodology provides new search variants (including region-based global goal computation), more extensive analysis of the proposed algorithms, and experiments using a service robot searching for objects in a domestic environment.

3 Methodology

3.1 A Bayesian Framework for Probability Map Updating

In this work, a spatial relation between two objects is defined as the probability distribution of the pose of the first object, π_A, given the known pose of the second object, π_B:

$$\mathrm{Rel}_{A,B}(\pi_A, \pi_B) = p(\pi_A|\pi_B) \tag{1}$$

Since the relation is treated as a probability distribution, the sum over all possible poses of A for a fixed pose of B is equal to 1:

$$\int_{\pi_A} p\,(\pi_A|\pi_B)d\pi_A = 1 \tag{2}$$

If the spatial relation depends only on the relative pose $\pi_{A/B}$, of object A with respect to object B, then the expression for the probability can be rewritten as:

$$Rel_{A/B}(\pi_{A/B}) = p\left(\pi_{A/B}\right) \tag{3}$$

In the most general case, the pose $\pi_{A/B}$ used in Eqs. 1 to 3 can be represented as a six-dimensional vector, with three components for position and three components for orientation. However, in this work the robot moves in a two-dimensional space parameterized by using coordinates (x, y).[1] Then, the relative pose is represented as a simple relative position of the object A described in the reference system of the object B: $\pi_{A/B} = (x, y)$. The space is quantized into square cells, each of size $k{\times}k$, and parameterized by using indexes (i, j). Then, the relative position can be represented as $\pi_{A/B} = (x, y) = (ki, kj)$. By using the index notation, a discretized spatial relation $R_{A/B}(i, j)$ can be constructed from the continuous spatial relation $Rel_{A/B}(x, y)$ by sampling over the grid:

$$R_{A/B}(i, j) = K_{norm} \cdot Rel_{A/B}(ki, kj) \tag{4}$$

$$\sum_i \sum_j R_{A/B}(i, j) = 1 \tag{5}$$

where K_{norm} is a normalizing constant. Note that Eqs. 4 and 5 are represented in the reference system of object B.

The term $p(a_{i,j})$ represents the probability that the center of the main object A is inside the space covered by the cell (i,j) of object's map a. Given a pose π_S of the robot's sensor, an observation z_A is performed inside a view cone dependent on π_S. The observation z_A consists of the result of executing a detector for object A over the current image, which can generate a positive detection $z_A = true$ when object A is found, or a negative detection $z_A = false$ when the object A is not found. The view cone corresponds to the set of cells observable from π_S. Both positive and negative observations z_A provide valuable information for the object search process, and can be used to compute an updated probability $p(a_{i,j}|z_A)$. The probability $p(a_0)$ is treated as a special case, and it represents the probability of the object being outside the search region. In this work, the search region consists of a single room. In the case of searching over multiple rooms, other complementary algorithms like Aydemir [25] can be used for selecting a room, and then the methodology proposed here can be used inside that room.

Positive detections $z_A = true$ provide information about the places where object A has a high probability of being, by means of a likelihood $p(z_A = true|a_{i,j})$, which is defined over the cells (i, j). This likelihood $p(z_A = true|a_{i,j})$ represents the probability of observing object A from robot pose π_S assuming that it is in a cell (i, j). When a cell is visible from pose π_S and the object is present in that cell, the likelihood of detecting the object in that cell is high. However, when the object is in a cell (i, j), positive object detections at other cells are necessarily false positives. Then, the likelihood of detecting the object in a cell not containing the object is low, and equal to the false positive rate of the object detector. Then, when the object is detected at a cell (i,j), the likelihood $p(z_A = true|a_{i,j})$ has a high value over the cell (i,j) where the object was detected, and a low value in the other cells. Negative detections $z_A = false$ provide information $p(z_A = false|a_{i,j})$ about the probability of not observing the object, assuming that it is in a cell (i, j). Then, cells inside the current view cone have a lower likelihood than the rest of the cells in the map.

The problem addressed in this work is for the robot to find a main object, A, by moving appropriately. The robot search process is applied until an instance of the main object A is found. In consequence, the search process includes only negative detections of the main object, A, before the object is found. An image I is captured at a sensor viewpoint π_S, then an object detector for object A is executed, and a detection z_A is obtained. Probabilities $p(a_{i,j})$ for cells inside the map, and $p(a_0)$ for the cell outside the map can be updated by using Bayes rule:

$$p\left(a_{i,j}|z_A = false\right) = \frac{p\left(z_A = false|a_{i,j}\right) p(a_{i,j})}{p(a_o) + \sum_{i,j} p\left(z_A = false|a_{i,j}\right) p(a_{i,j})} \tag{6}$$

$$p\left(a_0|z_A = false\right) = \frac{p(a_0)}{p(a_o) + \sum_{i,j} p\left(z_A = false|a_{i,j}\right) p(a_{i,j})} \tag{7}$$

When processing the image I, the object detector for the object B, can produce positive or negative detections z_B, which can be used to compute an updated probability $p(a_{i,j})$ for cells inside the map, and $p(a_0)$ for the cell outside the map. Then, a detection of object B can be used for updating the probability distribution for object A:

$$p\left(a_{i,j}|z_B\right) = \frac{p\left(z_B|a_{i,j}\right) p(a_{i,j})}{p(z_B|a_o)p(a_0) + \sum_{u,v} p\left(z_B|a_{u,v}\right) P(a_{u,v})} \tag{8}$$

$$p\left(a_0|z_B\right) = \frac{p(z_B|a_o)p(a_0)}{p(z_B|a_o)p(a_0) + \sum_{u,v} p\left(z_B|a_{u,v}\right) P(a_{u,v})} \tag{9}$$

The terms $p(z_B|a_{i,j})$ and $p(z_B|a_0)$ will be named cross-likelihoods, since they relate the detection of a secondary object, B, with the presence of the main object, A, on the map. These probabilities can be derived by considering

[1]The angular component of the robot's pose can also be used. However, in this work we choose to use only the position component of the pose.

probabilities $p(b_{u,v})$ for the presence of a secondary object, B, at cells (u, v) in the object's map b:[2]

$$p\left(z_B|a_{i,j}\right) = \sum_u \sum_v p\left(z_B|b_{u,v}\right) p\left(b_{u,v}|a_{i,j}\right) \tag{10}$$

$$p\left(z_B|a_0\right) = \sum_u \sum_v p\left(z_B|b_{u,v}\right) p\left(b_{u,v}|a_0\right) \tag{11}$$

The term $p(b_{uv}|a_0)$ is considered a constant over (u, v) whose sum has a value of 1 because an instance of B is supposed to be somewhere on the map. The term $p(b_{u,v}\ |a_{i,j})$ corresponds to the conditional distribution of B's position given A's position. Then, it corresponds also to the spatial relation between the main object, A, at location (i, j) and a secondary object, B, at location (u, v), as defined in Eq. 3, with the relative position between B and A being $\pi_{\mathrm{B/A}} = (i, j)\text{-}(u, v)$. By replacing this term with the spatial relation $R_{B/A}$, there is no need for storing a map for the secondary object; only the map for the main object and the likelihoods of the detections of the secondary object are needed:

$$p\left(z_B|a_{i,j}\right) = \sum_u \sum_v p\left(z_B|b_{u,v}\right) R_{B/A}\left(i-u, j-v\right) \tag{12}$$

$$p\left(z_B|a_0\right) = \frac{1}{n_U n_V} \sum_u \sum_v p(z_B|b_{u,v}) \tag{13}$$

where $n_{U\times}n_V$ is the size of the map.

Equation 12 can be implemented as a convolution in the (i, j) space between a likelihood image and a mask $R_{B/A}(i, j)$ describing the spatial relation between the main and secondary objects, which will be named a *spatial relation mask*:

$$p\left(z_B|a_{i,j}\right) = p\left(z_B|b_{i,j}\right) * R_{B/A}(i, j) \tag{14}$$

The proposed system is highly versatile because any translation-invariant spatial relation can be represented by an appropriate mask representing conditional distribution of B's position given A's position [1, 2]. The procedure used for updating the probability of the main object from detections of secondary objects in shown in Fig. 1.

It must be noted that extra secondary objects can be added to the system by creating additional spatial relation masks. In case these relations are chained, for example if object A is near B, and object B is near C, then the mask of the chained relation can be obtained by convolution of the original masks:

$$p\left(z_C|a_{i,j}\right) = p\left(z_C|b_{i,j}\right) * R_{B/A}(i, j) \tag{15}$$

$$p\left(z_C|a_{i,j}\right) = p\left(z_C|c_{i,j}\right) * R_{C/B}(i, j) * R_{B/A}(i, j) \tag{16}$$

$$\Rightarrow R_{C/A} = R_{C/B} * R_{B/A} \tag{17}$$

[2]Object's maps a and b are described in the same reference system, using the same grid resolution.

3.2 Creating Spatial Relation Masks from Co-occurrences

In this work, spatial relations are represented using masks, and then the system has a high flexibility for representing them. Moreover, complex spatial relations between objects can be also represented as linear combinations of basic spatial relations, as complex spatial relation masks can be represented as weighted sums of basic spatial relation masks. Each basic spatial relation corresponds to a semantic category meaningful to humans, i.e., to a qualitative spatial relation (QSR). In this work, we focus on four simple spatial relations: "very near" (VN), "near" (N), "far" (F), and "very far" (VF). The use of these spatial relations is appropriate as it enables to estimate a set of basic probability distributions from samples of relative positions of the objects in the real world. The masks for each of the spatial relations are defined in two versions: *hard masks* and *soft masks*. Hard masks are ring-shaped and are defined by two thresholds (a_1, a_2), and have a rectangular profile, while soft masks are also ring-shaped but are defined by four thresholds (a_1, a_2, a_3, a_4) and have a trapezoid-shaped profile, as shown in Fig. 2. Each basic mask is normalized to sum 1 over all of the cells on the map, thus a normalization constant is added to the formulas. Hard masks are defined as:

$$R_{B/A_{hard}}(i, j; a_1, a_2) = K * \begin{cases} 1 & a_1 \leq \sqrt{(ki)^2 + (kj)^2} < a_2 \\ 0 & other \end{cases} \tag{18}$$

while soft masks are defined as:

$$R_{B/A_{soft}} = K * \begin{cases} \frac{\sqrt{(ki)^2+(kj)^2}-a_1}{a_2-a_1} & a_1 \leq \sqrt{(ki)^2 + (kj)^2} < a_2 \\ 1 & a_2 \leq \sqrt{(ki)^2 + (kj)^2} < a_3 \\ \frac{a_4-\sqrt{(ki)^2+(kj)^2}}{a_4-a_3} & a_3 \leq \sqrt{(ki)^2 + (kj)^2} < a_4 \\ 0 & other \end{cases} \tag{19}$$

Soft and hard masks and their parameters are exemplified in Fig. 2.

In this work, basic masks are defined by a circle of radius u_1 in the case of "very near"; a circular ring of radii u_1 and u_2 in the case of "near"; a circular ring of radii u_2 and u_3 in the case of "far"; and a circular ring of internal radius u_3 with an external radius that covers the whole map in the case of "very far". Expressions for the hard basic masks are shown in Eqs. 20 to 23. Also, an extra gap parameter δ is considered when using soft masks for representing their soft

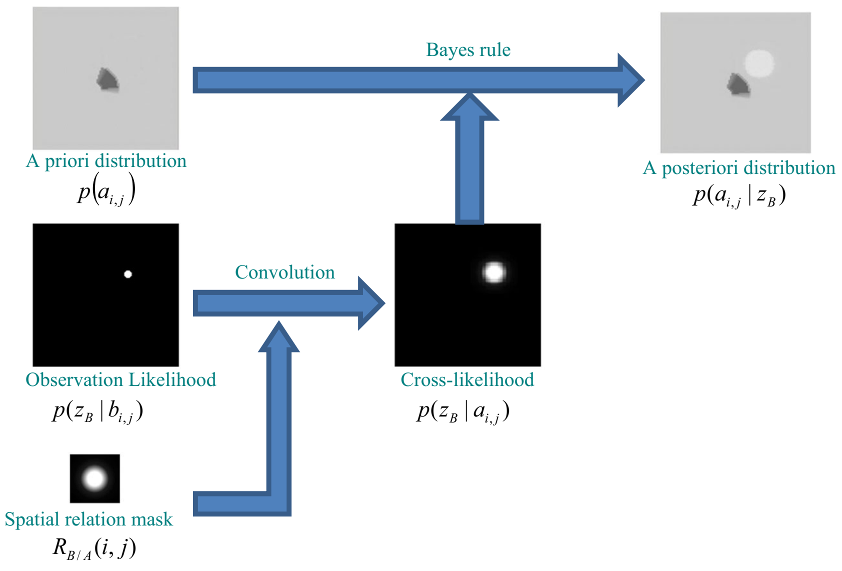


Fig. 1 Procedure used for computing the aposteriori distribution $p(a_{ij}|z_B)$ from the a-priori distribution $p(a_{ij})$, the observation likelihood of a secondary object $p(z_B|b_{ij})$ and the spatial relation mask $R_{B/A}(ij)$

borders. Expressions for soft basic masks are shown in Eqs. 24 to 27.

$$R^{VN}_{B/Ahard}(i,j) = R_{B/Ahard}(i,j;0,u_1) \quad (20)$$

$$R^{N}_{B/Ahard}(i,j) = R_{B/Ahard}(i,j;u_1,u_2) \quad (21)$$

$$R^{F}_{B/Ahard}(i,j) = R_{B/Ahard}(i,j;u_2,u_3) \quad (22)$$

$$R^{VF}_{B/Ahard}(i,j) = R_{B/Ahard}(i,j;u_3,\infty) \quad (23)$$

$$R^{VN}_{B/Asoft}(i,j) = R_{B/Asoft}(i,j;0,0,u_1-\delta,u_1+\delta) \quad (24)$$

$$R^{N}_{B/Asoft}(i,j) = R_{B/Asoft}(i,j;u_1-\delta,u_1+\delta,u_2-\delta,u_2+\delta) \quad (25)$$

$$R^{F}_{B/Asoft}(i,j) = R_{B/Asoft}(i,j;u_2-\delta,u_2+\delta,u_3-\delta,u_3+\delta) \quad (26)$$

$$R^{VF}_{B/Asoft}(i,j) = R_{B/Asoft}(i,j;u_3-\delta,u_3+\delta,\infty,\infty) \quad (27)$$

Radius selection is performed by ensuring that the obtained basic semantic masks are meaningful for the secondary objects involved in the object search, by selecting them to be discriminative for the secondary objects. Then, pairs of secondary objects are used for computing the radius. The radius values are selected by considering statistics of the distances d_{AB} and d_{AC} between two kinds of secondary objects B and C respect to the main object A, and by modeling their selection process as a classification problem. Thus, the optimal radius value between two categories, e.g., ”near” and ”far”, is the one that generates the same mean classification error in both classes B and C, when using the distances d from secondary objects to the main object A as the classification feature. This process is illustrated in a specific application in Section 4.4.

A complex mask can be created as a weighted sum of basic hard or soft masks:

$$R_{B/A}(x,y) = C^{VN}_{B/A}R^{VN}_{B/A}(x,y) + C^{N}_{B/A}R^{N}_{B/A}(x,y) + C^{F}_{B/A}R^{F}_{B/A}(x,y) + C^{VF}_{B/A}R^{VF}_{B/A}(x,y) \quad (28)$$

The four coefficients $C^{VN}_{B/A}$, $C^{N}_{B/A}$, $C^{F}_{B/A}$ and $C^{VF}_{B/A}$ are called co-occurrences because they indicate the relative frequency of occurrence of a pair of objects for each spatial relation. They can be constructed from samples of positions of both objects by computing the number of occurrences of each basic spatial relation. If a set of samples is divided into basic semantic categories, and the count is n_{VN} for ”very

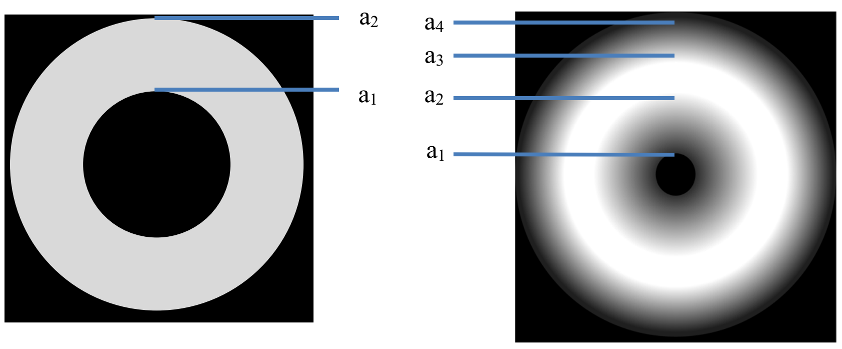


Fig. 2 Basic hard mask (left), basic soft mask (right) and the parameters determining their shape

near”,n_N for “near”, n_F for “far”, and n_{VF} for ”very far”, the co-occurrences can be computed as:

$$C^{VN}_{B/A} = \frac{n_{VN}}{n_{VN}+n_N+n_F+n_{VF}} \tag{29}$$

$$C^{N}_{B/A} = \frac{n_{N}}{n_{VN}+n_N+n_F+n_{VF}} \tag{30}$$

$$C^{F}_{B/A} = \frac{n_{F}}{n_{VN}+n_N+n_F+n_{VF}} \tag{31}$$

$$C^{VF}_{B/A} = \frac{n_{VF}}{n_{VN}+n_N+n_F+n_{VF}} \tag{32}$$

3.3 Search Strategy

The object search methodology depends on the object search strategy, but also on the selected world representation. In this work, the world is represented as a grid map, with one additional cell for representing the case of objects outside the map. The probability distribution of the position of the searched object on the grid map is named a probability map. The spatial relation between the main object and a secondary object is represented by combining basic spatial relation masks into a complex spatial relation mask, which can be a hard or a soft relation mask. One or several secondary objects can be used. Object detections, obtained using the L&R SIFT object detector [26–28], can be positive (object detected) or negative (object not detected). Both kinds of detections can be considered for updating the probability map. Also, an occupancy map is used for planning and for avoiding collisions, which is named obstacle map.

The search is performed by selecting an optimal robot pose to be reached and executing a planner for generating the precise path. Different methodologies for generating optimal poses to be reached generate different object search strategies, as they minimize different cost functions defined by an expected utility, which could consider both the probability map and the distance between the endpoints of the trajectory.

In this work, two search strategies are considered: local search, and global search using regions. In addition, a third strategy that uses probabilistic object-object relations is described in Section 3.3.3, because it is used by the work proposed by Elfring et al. [16], which is used in the comparisons presented in Sections 4 and 5.

The proposed local and global search strategies have flavors: (i) they can use soft or hard relation masks, and (ii) they can use only positive or both positive and negative information from detections of secondary objects. The search procedures are able to perform detection and planning while the robot is moving, then it is not necessary for the robot to reach a goal before re-planning. When an obstacle map is available, all of the methods are able to use it for generating only valid trajectories. Also, the use of global search using regions is a novel, fast procedure for selecting the goal with the maximum expected utility overall the map.

3.3.1 Local Search

A path for searching for a given object can be created by generating an optimal viewpoint at each iteration. The optimal viewpoint is generated from a set of random sensor poses reachable within a fixed time, with the one that maximizes the probability of finding the object in the visible area selected [8] as:

$$\underset{k=1..N}{\arg\max} \sum_{i=1}^{n}\sum_{j=1}^{n} p(a_{i,j})V(a_{i,j},k) \tag{33}$$

where N is the number of candidate poses, and $V(a_{i,j},k)$ is defined as:

$$V(a_{i,j},k) = \begin{cases} 1, & \text{if } a_{i,j} \text{ is inside the k}^{th} \text{ view cone} \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

A new goal pose is recomputed periodically every S seconds, or when the robot reaches the current goal. Plan reevaluation enables the robot to use the latest object detections for the plan, without needing the robot to stop for detecting objects. In the experiments reported in Sections 4 and 5 we chose to use a simple search strategy in which the robot, at each iteration, selects the execution of a composed sequence of movements consisting of one initial rotation, followed by a translation and a final rotation (N =3). The parameter’s space associated with this sequence of movements is explored randomly, and the best sequence is chosen as the one that maximizes the probability of finding the searched object. When an obstacle map is available, the movements that cause collision between the robot and the obstacles are not considered to be valid solutions. Algorithm 1 shows the proposed local object search procedure.

3.3.2 Global Search using Regions

The probability map is divided into regions by using a coarser grid over the probability map grid. Each of the regions covers a rectangular area in the probability map. Each region has a probability that is equal to the sum of the probabilities of the cells covered by the region. Also, each region has a centroid that is located at the center of the rectangle. In our settings, a region covers 5×5 cells from the probability map.

When an obstacle map is provided, it is dilated for ensuring that only traversable paths can be considered when using the map. Sometimes, the centroid of a region can be inside a non-traversable section of the map. In that case, the nearest free cell is used for generating a corrected centroid for the

Algorithm 1 Proposed Informed Local Search Procedure. It uses Eqs. 6–9, 12–13, 20–28, 33 and 34

```
1.  Initialize probabilities and masks (Eqs. 20 to 28)
2.  While the main object is not detected:
3.      Run a detector for the main object
4.      If the main object is detected, the algorithm finishes
5.      Update the probability map of the main object (Eqs. 6 and 7)
6.      Run a detector for each secondary object
7.      For each detected secondary object
8.          Compute the cross-likelihood for positive detections (Eqs. 12 and 13)
9.          Update the probability map (Eqs. 8 and 9)
10.     For each non-detected secondary object
11.         Compute the cross-likelihood for negative detections (Eqs. 12 and 13)
12.         Update the probability map (Eqs. 8 and 9)
13.     Every S seconds, compute a new goal by selecting the optimal viewpoint achievable in a given time. The optimal viewpoint is the one that has the largest probability of containing the main object inside its view cone and that does not cause collisions (Eqs. 33 and 34)
14.     Go to 2
```

region. If the corrected centroid is outside the region, then that region is discarded.

The robot must select an optimal region to be reached. Then, the expected utility (EU) is computed for each of the regions. The formulation of the EU is based on [16]. The EU of a region depends on both the probability of the region, and on the distance between the current position of the robot and the centroid of the region. When an obstacle map is available, the distance is computed over the shortest valid path that connects the robot and the centroid of the region, by using a theta-star algorithm [29]. As the original obstacle map could have a high resolution, it must be downsampled before applying the theta-star algorithm for enabling real-time shortest path computations. For facilitating computations, the obstacle map is downsampled to the same resolution used for the probability map. In each frame, the expected utility for a region is computed as:

$$EU(R, \pi_R) = P_{reg}(R) + w * \frac{1}{\text{atan}\,(dist(\pi_R, R))} \tag{35}$$

where R is the region, $\pi_R$is the pose of the robot, $P_{reg}(R)$ is the sum of the probabilities inside the region R, $dist(\pi_R, R)$ corresponds to the length (in meters) of the shortest valid path between the robot and the centroid of the region, and w is a weight that balances distance and probability. The region with the largest EU is selected as the navigation target, and the last pose of the path (that is inside the region) is selected as the navigation goal. When the robot is near the centroid of a region, the region is observed and then considered to have been visited, and then it is deleted from the region grid. Algorithm 2 summarizes the global search procedure.

Algorithm 2 Proposed Informed Global Search Procedure. It uses Eqs. 6–9, 12–13, 20–28 and 35

```
1.  Initialize probabilities and masks (Eqs. 20 to 28)
2.  While the main object is not detected:
3.      Run a detector for the main object
4.      If the main object is detected, the algorithm finishes
5.      Update the probability map of the main object (Eqs. 6 and 7)
6.      Run a detector for each secondary object
7.      For each detected secondary object
8.          Compute the cross-likelihood for positive detections (Eqs. 12 and 13)
9.          Update the probability map (Eqs. 8 and 9)
10.     For each non-detected secondary object
11.         Compute the cross-likelihood for negative detections (Eqs. 12 and 13)
12.         Update the probability map (Eqs. 8 and 9)
13.     Every S seconds, compute a new goal by selecting an optimal pose to be reached. The optimal pose is defined as the last pose in the path reaching the centroid of the valid region with the largest expected utility (EU) (Eq. 35).
14. Go to 2
```

3.3.3 Search Exploiting Probabilistic Object-Object Relations

In this search algorithm, described in Elfring et al. [16], a probability map is not used. Instead, probabilities $p(x_i|x_j)$ are computed by counting co-occurrences of x_i and x_j in the same image. The algorithm works by updating several lists of objects: X is the set containing the object's instances observed by the robot; F is the set containing objects marked as dead end; and T is the set containing the (intermediate) target objects (see the search procedure in Algorithm 3). For selecting the optimal target object, an expected utility is computed as:

$$EU(o, x_t) = p(o|x_t) + w * \frac{1}{\text{atan}\,(dist(o, x_t))} \tag{36}$$

where x_t is the object to be found, and o an object in X.

Algorithm 3 Informed Search Procedure from Elfring et al. [16]

```
1. Initialize $F=Ø$, T ={$x_t$}
2. While the main object is not detected:
3.     Run the object detectors and store their detections in X
4.     $X = X \setminus F$, then determine the pair $o^*$ from X and $x^*$ from T that maximizes $EU(o^*, x^*)$ in Eq. 36
5.     If $EU(o^*, x^*) \geq$ threshold, start navigating towards object $o^*$
6. If $EU(o^*, x^*) <$ threshold, the observed objects have no sufficient relation with the target object and an intermediate object needs to be added. The object $x_s$ with the strongest relation to any of the objects in T according to Bayes' law is added to T: $T := T \cup \{x_s\}$. With T updated, go to step 3
7.     If the robot reaches an intermediate object, it is added to F since it did not lead to the target object. Now go back to step 3 and try to find a route via another object. If such a route is not available, start a random search and update X if new objects are found.
8.     Go to 2
```

4 Simulation Experiments

4.1 Experimental Setup

In order to characterize and validate the proposed object search methodology, a simulation environment was developed. This environment is based on the use of the *Player/Stage* simulator [30], but it incorporates: (i) statistics about the co-occurrence of the object's classes in the real-world to compute the parameters of the spatial relation masks, (ii) a model for the observation of the objects in 3D by the robot, and (iii) the use of an object detection simulation system in 3D, whose performance depends on the relative pose between the object and the robot's camera, as well as the object class. Note that motion of the robot in Player/stage is 2D, but object detection simulation is performed in 3D.

The simulation environment was developed in order to obtain replicable experiments and a better characterization of the proposed object search methodology.

4.2 Modeling the 3D Object's Observation

Player/Stage is designed to manage 2D environments. Its fiducial detector defines a 2D field of view, and a minimum and a maximum object detection range. However, for modeling the observation of objects in a 3D environment, additional parameters need to be included, like the pose of the robot's camera, the height of the object, and its size in 3D. Figure 3 shows the robot and camera configuration used in this work.

For modeling/simulating the observation obtained by a robot in a 3D environment, the following transformation matrix must be used:

$$\begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} = \begin{bmatrix} \cos(-\varphi) & 0 & -\sin(-\varphi) \\ 0 & -1 & 0 \\ \sin(-\varphi) & 0 & \cos(-\varphi) \end{bmatrix} * \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_{obj} - x_r \\ y_{obj} - y_r \\ z_{obj} - z_r \end{bmatrix} \tag{37}$$

where φ is the pitch angle of the camera, θ is the yaw angle of the robot on the horizontal axis; $(x_{obj}, y_{obj}, z_{obj})$ is the position of the center of the object; (x_r, y_r, z_r) is the position of the robot's camera; and $(x_{cam}, y_{cam}, z_{cam})$ is the position of the object in the camera system. Then, the relative pose of the object is computed as:

$$r_d = \sqrt{x_{cam}^2 + y_{cam}^2 + z_{cam}^2} \tag{38}$$

$$\theta_d = \text{atan2}(y_{cam}, x_{cam}) \tag{39}$$

$$\varphi_d = \text{atan2}(z_{cam}, x_{cam}) \tag{40}$$

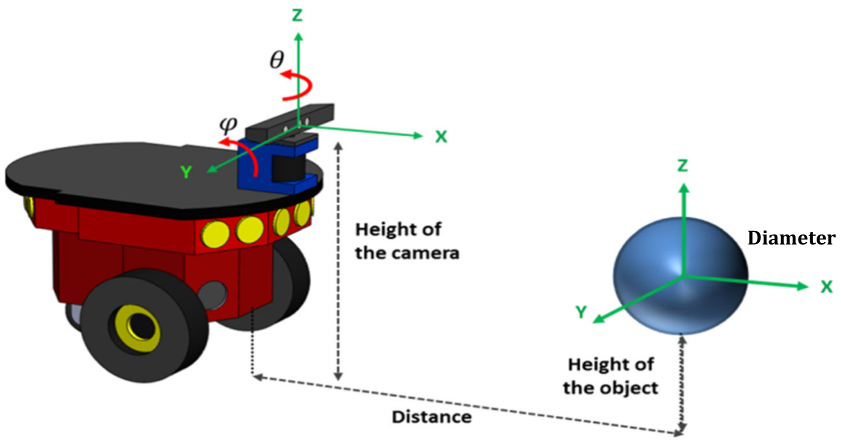
with r_d the distance from the object to the camera, θ_d the yaw angle of the object in the camera system, and φ_d the pitch angle of the object in the camera system. A 3D object can be detected if its center is contained in the robot's field of view defined as: $r_d \in [r_{min}, r_{max}]$, $\theta_d \in [\theta_{min}, \theta_{max}]$ and $\varphi_d \in [\varphi_{min}, \varphi_{max}]$.

The simulation of occlusions between objects in a 2D environment is different from that in the 3D case (see Fig. 4). In the 3D case, we model each object as a sphere whose center is placed at the center of the object, and whose diameter ρ depends on the object size. Each pair of spheres contained in the field of view is projected in the image plane and whether or not they intersect in the image space is verified. In case they intersect, an occlusion has been detected, and only the nearest object is selected as visible. This procedure is shown in Algorithm 4.

4.3 Object Detection Simulation

In order to simulate the detection of objects, statistics were calculated on the performance of a specific object detection system under different camera-object distances and angles. This data was used to build an object-detection LUT (Look-Up Table) that stores the detection rates, which are used by the simulator every time the robot tries to detect an object placed inside its field of view. Then, when the robot observes an object inside the simulator, the pose of the object respect to the robot is used for retrieving a detection rate from the LUT. A random number between 0 and 1 is

Fig. 3 Geometric parameters needed for simulating 3D detections and determining occlusions The object is modeled as a sphere projected on a pin-hole camera. The parameters related to the pose of the sensor respect to the robot (height of the camera), the position of the object respect to the robot (distance, height of the object) and the diameter ρ of the sphere need to be estimated



Algorithm 4 Procedure for Computing Occlusion of Objects.

```
1.  For each pair of detected objects:
2.      Object 1 is described by: position $x_1, y_1, z_1$, diameter $\rho_1$
3.      Object 2 is described by: position $x_2, y_2, z_2$, diameter $\rho_2$
4.      $ang\_proy_1 = atan2(\rho_1, x_1)$
5.      $\theta_1 = atan2(y_1, x_1)$
6.      $\varphi_1 = atan2(z_1, x_1)$
7.      $ang\_proy_2 = atan2(\rho_2, x_2)$
8.      $\theta_2 = atan2(y_2, x_2)$
9.      $\varphi_2 = atan2(z_2, x_2)$
10.     $dist = \sqrt{(\theta_2 - \theta_1)^2 + (\varphi_1 - \varphi_2)^2}$
11.     If $dist > ang\_proj_1 + ang\_proy_2$
12.             Both $obj_1$ and obj$_2$ can be detected
13.     Else
14.             Occlusion, the farthest object cannot be detected
```

generated. If the random number is lower than the detection rate, a detection of the object is generated.

For computing the detection rates to be stored in the LUT, an object detection system that uses HOG descriptors as features and SVMs as classifiers was used in a real-world setup. A different SVM is used for each object class and object orientation. Each SVM is trained by computing HOG descriptors from five sets of 50 labeled images that show the object under the required orientation for 5 different distances, using the setup described in Fig. 5. The SVM training procedure is described in [31].

The object's detection statistics are computed using images captured from 40 different viewpoints. As shown in Fig. 5, the viewpoints correspond to camera-object distances of 80, 120, 160, 200 and 250 centimeters, and camera-object yaw angles of 0, 45, 90, 135, 180, 225, 270, and 315 degrees (pitch and roll camera-angles are not considered). For each viewpoint, 100 images of each object are captured by moving the camera while keeping it in a pose compatible with the current viewpoint. In total 4,000 images of each object are obtained. A detection probability is computed by counting the number of detections in the 100 available images for each viewpoint and each object. Four objects are considered: "monitor," "keyboard," "system unit," and "router". The detection probabilities are summarized in Table 1, and coded into the object-detection LUT.

4.4 Creation of Co-occurrence Matrices

A set containing a total of 243 manually labeled images obtained from both the LabelMe [14] and the Flickr websites was used for generating co-occurrence matrices. In each image, instances of the objects, "monitor," "system unit," "keyboard," and "router", were labeled by manually selecting four corners on the image. Since the sizes of the objects and the parameters of the camera were known, it was possible to estimate the position of each object in space by using a homography. This procedure was validated by using a Kinect camera, using 31 monitor images, and 25 keyboard images. The errors obtained when predicting the depth from the homography were all below 12 centimeters, and then lower than the thresholds used for creating the

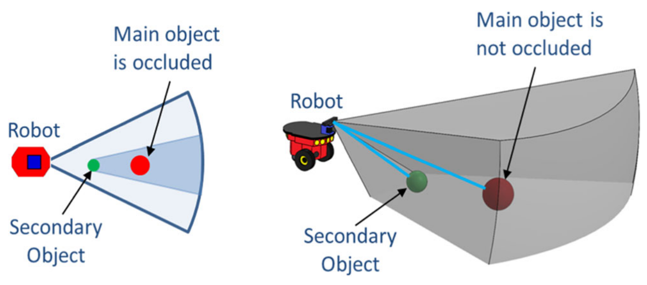


Fig. 4 Visualization of Occlusion in a 2D (left) that is not a real occlusion in 3D (right)

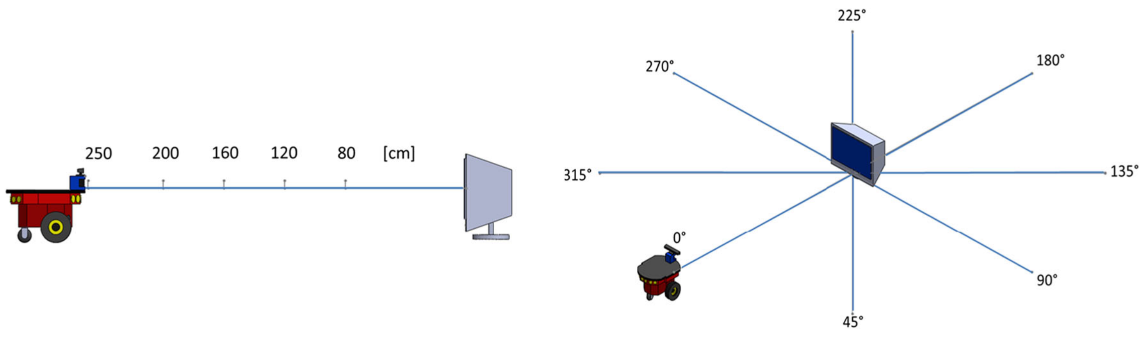


Fig. 5 Viewpoints with different distances and different angles

spatial relation masks. Also, as object instances are manually labeled, images containing small objects that are far between can be used. Several instances of the objects on the set of images and their poses were used to construct co-occurrence matrices for the categories "very near," "near," "far," and "very far" for each of the objects with respect to the others. As described in Section 3.2, the radius values are selected by considering statistics of the distances between two kinds of secondary objects respect to the main object, and by modeling their selection process as a classification problem. Thus, the optimal radius value between two categories, e.g., "near" and "far", is the one that generates the same mean classification error in both classes when using only the distances from secondary objects to the main object for classifying the secondary objects. For the radius that separate "very near" from "near", distances keyboard-monitor v/s cpu-monitor are considered. For the radius that separate "near" from "far", cpu-monitor v/s router-monitor are considered. Finally, the radius that separate "far" from "very far" is the maximum distance from monitors (the main object) to routers in the available database. Note that the threshold values used for delimiting the spatial relations are computed automatically from collected real-world data, by trying to maximize the discriminative/descriptive power of the resulting spatial relations. Then, the procedure for determining the thresholds is data-driven, but computed offline.

Given the poses of a pair of objects, a distance was computed and used for selecting whether the sample belongs to the categories "very near," "near," or "far". If an object is detected alone in an image, the sample belongs to the category "very far".[3] Only the depth and horizontal axis were used to compute the distances, since differences in the vertical direction do not affect the position of the object when it is transferred onto the 2D grid. As an example, the final co-occurrences for the object "monitor," as the main object, are shown in Table 2.

The basic hard and soft masks are defined by Eqs. 18–22 and 23–27, respectively. The thresholds that separate the categories, "very near," "near," "far," and "very far" are $u_1 = 60$[cm], $u_2 = 100$[cm], and $u_3 = 150$[cm]. The gap parameter for the soft masks is $d = 20$[cm].

4.5 Experiments

Experiments were performed on maps containing four objects. Each map contains a main object, named A (monitor), to be searched for, and three secondary objects, named B (keyboard), C (system unit), and D (router). The size of each map is 6[mt] x 6[mt]. A total of 20 maps was created by picking a random position for the main object, A, and then picking random positions for the objects B, C, and D following a distribution that represents their co-occurrences. Exemplar configurations are shown in Fig. 6.

Objects were placed at 0.5[mt] from the floor, and the height of the camera is 1[mt]. The orientations of the objects in each map were selected randomly as multiples of 45 degrees. The 20 maps were used to perform the experiments, each map being used the same number of times as the others. A laser sensor from *Player/Stage* was used for avoiding collisions. Observations of the objects were obtained by using a semantic camera, i.e. the 3D observation system described in Section 4.2. The field of view of the sensor has a horizontal range of 75 degrees, a vertical range of 45.6 degrees, a minimum depth of 0.3[mt], and a maximum depth of 2.5[mt]. The probability map was initialized with a uniform distribution. It has to be noted that, while player/stage is 2D, the detection of objects is simulated in 3D space, considering possible occlusions between objects. The simulated semantic camera can run at 3 fps, considering that the SIFT detector used in the real robot is able to run at that speed.

In each experiment the goal is to find the main object A in less than 5 minutes. In the search process, only negative detections of the main object need to be processed because a positive detection causes the search process to end. Eleven

[3]Objects that are normally very far from each other usually do not appear in the same image. Therefore, instead of using the term "uncorrelated" for this kind of objects, we prefer to use the category "very far".

Table 1 Object detection Probability (%) for: (a) System Unit, (b) Monitor, (c) Router and (d) Keyboard

% Det	System Unit: Distance [cm]					Monitor: Distance [cm]					Router: Distance [cm]					Keyboard: Distance [cm]				
Angles	80	120	160	200	250	80	120	160	200	250	80	120	160	200	250	80	120	160	200	250
0°	81	76	69	42	21	33	40	57	67	84	51	67	54	32	0	34	62	80	64	59
45°	15	31	45	39	22	12	38	40	85	89	24	39	62	41	17	52	85	82	79	45
90°	66	76	80	69	29	49	57	67	76	81	56	65	47	29	0	5	28	26	64	79
135°	12	35	49	31	24	17	75	70	76	74	52	49	39	33	15	36	80	42	38	36
180°	88	94	78	55	26	45	48	58	59	67	43	75	54	27	0	41	43	64	50	33
225°	11	39	40	35	19	07	95	74	78	72	48	46	37	33	13	60	82	86	63	52
270°	68	78	82	69	46	38	55	64	71	84	55	57	45	34	0	06	15	25	63	75
315°	20	32	44	38	25	10	56	45	65	60	47	58	48	43	14	44	86	62	58	68

different object search methods are tested on the available set of maps: a baseline method (UIS) which uses direct search instead of informed search; the method proposed by Aydemir [8] (ISUP); the method proposed by Elfring [16] (AOSEOOR); and eight different variants of the proposed methodology which use local search or global search (ISC:Informed Search using Convolutions versus ISCG: ISC Global), hard or soft masks (HM: Hard Masks versus SM: Soft Masks), and positive and negative information on the detection of secondary objects, or just positive information (PN: Positive&Negative versus OP: Only Positive). The methods are the following:

1. UIS: Uninformed/direct local search (the baseline method). A probability map $p(a_{i,j})$ for object A is estimated by using negative detections of that object, and then object A is searched for by finding viewpoints that maximize the probability of containing A.
2. ISUP: Informed search using particles. The method of Aydemir [8] is used for constructing a probability map $p(a_{i,j})$ for object A by using negative detections of that object. Then object A is searched for by finding near viewpoints that maximize the probability of containing A. When a secondary object, B, is detected, a set of 500 particles is generated around the detection inside the current view cone. Each particle represents a hypothesis about the location of object A, given the known position of object B in the map. The spatial relation between objects A and B induces a probability distribution of the presence of A, given the position of B, that has a maximum value over the map. The particles with associated probability value of equal, or greater than half of the maximum probability are used to select the next optimal viewpoint by maximizing the number of accepted particles inside the view cone.
3. AOSEOOR: Active object search exploiting object-object relations: The method of Elfring [16] is used for creating lists of observed objects. The target object with the highest expected utility is selected for generating the next navigation goal. The search procedure described in Algorithm 3 is used.
4. ISC-PN-HM: Informed Search using Convolutions with positive and negative information and hard relation masks. A probability map $p(a_{i,j})$ for the main object A is estimated by using positive and negative detections of objects B, C, and D, negative detections of object A, and hard spatial relation masks. Then object A is searched for by finding near viewpoints that maximize the probability of containing it. The local search procedure described in Algorithm 1 is used.
5. ISC-OP-HM: Informed Search using Convolutions with only positive information and hard relation masks: Similar to method 4, but in this case only positive detections of objects B, C, and D are used.
6. ISC-PN-SM: Informed Search using Convolutions with positive and negative information and soft relation masks. Similar to method 4, but in this case soft spatial relation masks are used.
7. ISC-OP-SM: Informed Search using Convolutions with only positive information and soft relation masks. Similar to method 5, but in this case soft spatial relation masks are used.
8. ISCG-PN-HM: Informed Global Search using Convolutions with positive and negative information and hard relation masks: similar to method 4, but in this

Table 2 Final co-occurrences of objects around the object "monitor"

Semantic categories	Secondary objects		
	Keyboard	System unit	Router
Very Near	0.773	0.178	0.061
Near	0.143	0.491	0.151
Far	0.046	0.258	0.485
Very Far	0.038	0.074	0.303

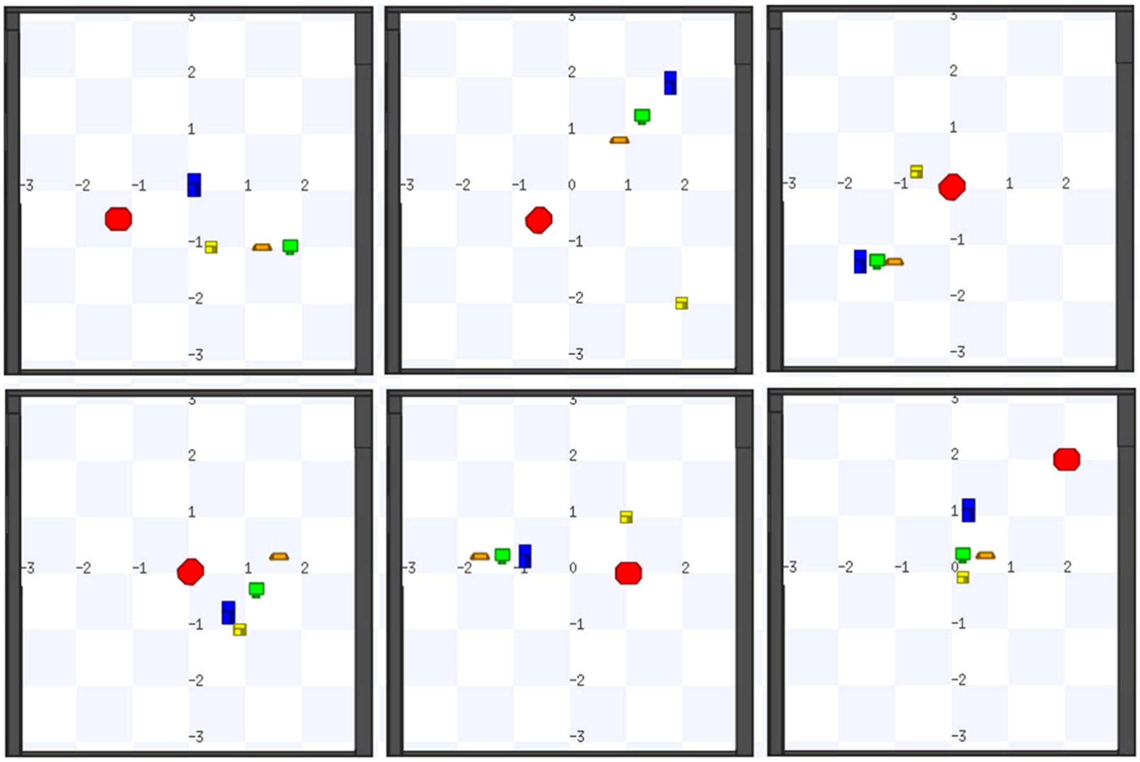


Fig. 6 Examples of initial configurations of objects following the estimated spatial relations. The robot is represented in red, the main object (CPU) is represented in blue, and secondary objects are represented in green (monitor), orange (keyboard) and yellow (router)

case region-based global search is used. The global search procedure described in Algorithm 2 is used.

9. ISCG-OP-HM: Informed Global Search using Convolutions with only positive information and hard relation masks: similar to method 5, but in this case region-based global search is used.
10. ISCG-PN-SM: Informed Global Search using Convolutions with positive and negative information and soft relation masks: similar to method 6, but in this case region-based global search is used.
11. ISCG-OP-SM: Informed Global Search using Convolutions with only positive information and soft relation masks: similar to method 7, but in this case region-based global search is used.

In all of the methods that require computing an EU (methods 3, 8, 9, 10, 11), the weight parameter for the EU is set to $w = 0.1$ in Eqs. 35 and 36.

In addition, different view scales are tested. A view scale corresponds to the maximum distance in meters at which the main object can be detected. In Table 3 this maximum distance varies between 1 and 2.5 meters. The secondary objects can always be detected at a fixed distance of 2.5 meters. 200 experimental trials were performed for each method and view scale. Then, a total of 15,400 experimental instances were executed for characterizing the performance of the methods. It must be noted that in all view scales, the same distance thresholds for creating the masks are used, i.e., they are not dependent on the view scale.

The results for the object detection experiments for different view scales are shown in Table 3.

Table 3 shows clearly that the methods using informed search outperform the baseline method that uses uninformed/direct search (UIS). It can also be seen that six of the indirect search methods built using the proposed methodology obtain better performance than those of Aydemir (ISUP) and Elfring (AOSEOOR) for all view scales. In fact, ISCG-PN-SM obtains the best performance; its detection rate is much higher than those of ISUP and AOSEOOR (at a scale of 1 meter: 79% versus 31% and 51%; at a scale of 2.5 meters: 95% versus 70% and 83%). The proposed methods perform better because they have the ability to store detections of secondary objects in the probability map. On the other hand, ISUP is reactive in the sense that it requires the secondary object to be observed in the current frame for hypothesizing the pose of the main object. AOSEOOR does not use a probability map, then it behaves randomly before it detects a secondary object. Then, the proposed methods are the only able to store all of the information about positive and negative detections of all of the objects, and use them for planning paths.

The results also show that (i) the use of convolutions with soft masks outperforms the use of hard masks, (ii) the use

Table 3 Search success rate (%) for several view scales (maximum detection distance of the sensor for the main object) for each of the tested methods in simulations

Methods	View scale in meters						
	1.0	1.25	1.5	1.75	2.0	2.25	2.5
UIS	28	26	38	50	58	65	65
ISUP	31	38	46	53	67	70	70
AOSEOOR	51	63	71	76	73	78	83
ISC-PN-HM	49	60	63	64	77	79	85
ISC-OP-HM	41	42	44	44	59	65	67
ISC-PN-SM	53	64	68	71	77	80	88
ISC-OP-SM	42	42	48	55	62	67	69
ISCG-PN-HM	77	75	77	87	89	90	93
ISCG-OP-HM	70	70	76	84	84	87	89
ISCG-PN-SM	79	74	86	87	88	94	95
ISCG-OP-SM	74	76	82	82	87	88	91

of both positive and negative information of the detection of secondary objects is better than using just the positive information, and (iii) the use of global search improves the search success rate of the methods, and it is able to work even with small view scales. The fact that soft masks work better is explained because hard masks generate a rough quantization of the spatial relations, and cause the probability map to be pixelated. A pixelated map causes zones that are near to have very different values when evaluating an optimal viewpoint, then obtaining worse results than when using a softer map. The fact that using both positive and negative information in simulations improves the success rate is explained because the non-detection of secondary objects gives information about absence of the main object in the current view cone. However, use of negative information requires having a good observation model. The fact that global search improves the results occurs because local search strategies could be trapped in areas with obstacles, and only a global search is able to generate new views outside the trapping zone.

Finally, it can be observed that the search success rate of all methods increases with the scale (i.e. distance) at which the objects can be detected. This is to be expected since if the objects can be detected at a greater distance, then the rate of the search process improves. Also, results show that object search procedures are stable against variations in the size of the objects.

4.6 Scaling of Processing Time with the Number of Objects

The processing time of the proposed informed search algorithms can be optimized by observing that the cross-likelihood images are constant, except around the detection area in the case of positive detections, and around the view cone in the case of negative detections. This occurs because the spatial relation between two objects is constant when the distance between them is large enough. Considering that the computation of convolutions in constant value areas is not needed, focalized convolutions can be applied only in the non-constant areas. The non-constant region inside the cross-likelihood $p(z_B|b_{i,j})$ is denoted by L, and its size (l_x, l_y). If the relation mask has a size (m_x, m_y), then the region L must be enlarged for enabling a full computation of the convolution with the mask. Then, the required number of multiplications n_{mul} is quadratic respect to the size of the mask:

$$n_{mul} = (l_x + m_x) * (l_y + m_y) * m_x * m_y \tag{41}$$

The number of multiplications in the convolution is not dependent on the full size of the map (i_x, i_y) because the convolution is focalized around the detection, and then large maps could be used. Note also that the number of multiplications n_{mul} in Eq. 41 is bounded by the convolution in Eq. 14. In other words, if computing (14) is faster than computing (41) because of a very small map, then (14) can be used instead.

When using both positive and negative information, the computation of the cross-likelihoods is required for each of the objects, even if they are not detected, and then the processing time (PT) increases linearly with the number of secondary objects. However, if only positive detections are processed, the number of cross-likelihoods to be computed depends on the number of currently detected secondary objects, and then the required processing time remains almost constant.

In the performed experiments, a small map (100x100) and mask (32x32) were used for representing a 6[mt] x 6[mt] environment. In this case, the convolutions can be computed quickly and the PT of the informed search process is lower than the PT of the object detection process.

5 Real Experiments Results

5.1 Experimental Setup

Real world experiments using a service robot (Bender [32, 33]; see Fig. 7) and a domestic home-like environment were carried out. The robot, which has participated in several service robotics competitions (e.g. RoboCup@Home), is equipped with a Pioneer base, two Hokuyo laser sensors, an RGB-D camera (ASUS sensor), a tablet computer which provides a user-friendly interface, and an Alienware notebook used for running computer vision algorithms. The experiments were carried out in a domestic home-like environment containing walls, furniture (e.g. tables) and several objects normally found in domestic environments. The environment has a size of 6.6[mt] x 9.4[mt], and its layout is shown in Fig. 8.

In the reported search experiments the pose of the robot is not known in advance, and the robot needs to self-localize. The robot is provided with an obstacle map of the environment, and standard ROS-based modules for self-localization (amcl) and navigation (navfn + base local planner) are used. Objects are detected by using the L&R SIFT object detector [26–28] with its parameters optimized for this particular problem. In Fig. 9 is shown an example of a pair of detected objects.

5.2 Experiments

Seven different object search methods were tested: the baseline direct search method (UIS), the method proposed by Aydemir [8] (ISUP), the method proposed by Elfring [16] (AOSEOOR), and four different variants of the proposed methodology. We did not consider the use of hard masks because of their consistently lower performance in the simulation experiments.

The environment contains two surrogate objects, A (main) and B (secondary), which have the same spatial relation as the "monitor" and "keyboard" objects, but are easily detectable by using the L&R SIFT object detector [26–28], that is able to find object instances of each specific object. Then, the same relation masks used for the simulated experiments are also used for the real ones (same thresholds for separating the semantic categories "very near," "near," "far" and "very far"). Objects are placed over a desk, with their positions according the spatial relations estimated before. Detections are performed by using the RGB-D camera (ASUS sensor). The positions of the detected objects

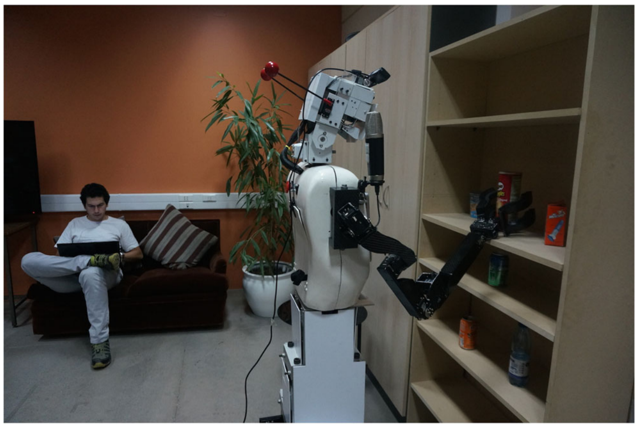


Fig. 7 Robot (Bender) used in the real world experiments

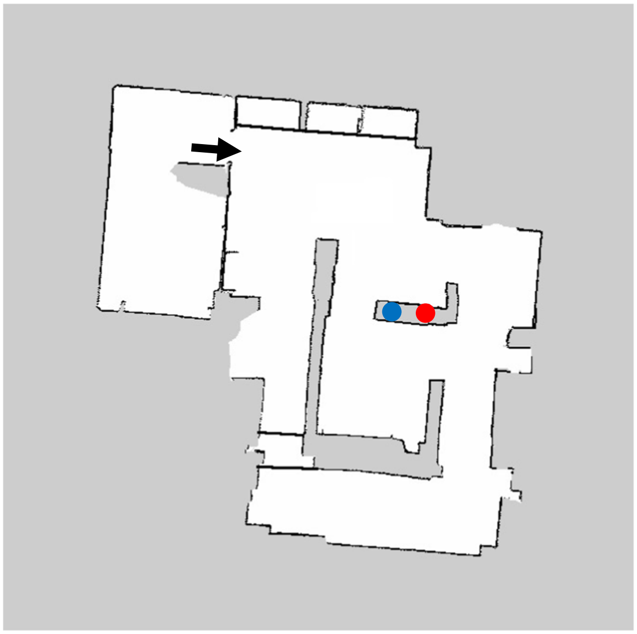


Fig. 8 Layout used for the Real World Experiments. As an example of a possible configuration of the robot and the objects, the initial robot position is illustrated by the arrow, and the placements of the main and secondary objects are shown by the red and blue circles respectively. Note that this map is built by using a LIDAR sensor placed parallel to the ground, at 28 cm above the ground level. Then, furniture like tables (covered by tablecloths) is also represented as gray areas

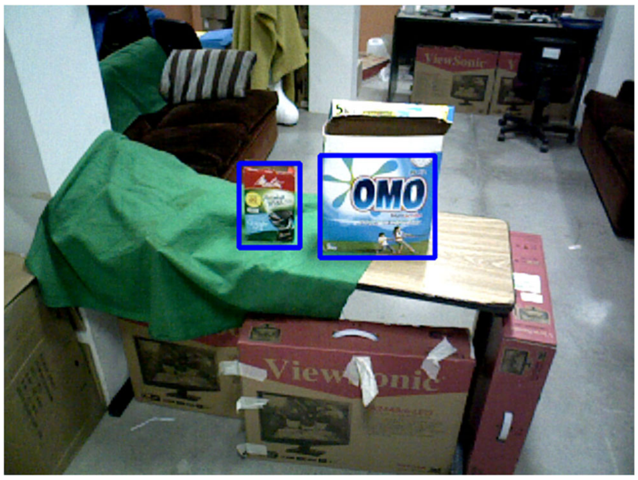


Fig. 9 Objects detected by using the L&R SIFT detector

with respect to the robot base can be estimated reliably using the ASUS range data. The maximum distance at which the objects can be recognized depends on the SIFT recognizer, i.e., no hard-coded maximum detection distance is used. However, empirical maximum detection distances are close to the ones used in simulations (1 meter for the main object, 2.5 meters for the secondary object) because of the selected sizes of the objects (the main object is smaller than the secondary ones).

The probability map was initialized with a uniform distribution. The task to perform consists of finding the main object in the room within a maximum allowable searching time of 5 minutes. Each method is able to re-plan a new goal for the navigation module every 6 seconds. Also, object detection is performed while the robot is moving. For each method, 15 instances of the experiment are performed (Set 1). The results of the experiments are shown in Table 4 (right column) in terms of the detection rate (%). For comparative purposes, the detection rates obtained in simulations, at a similar view scale, are also shown (left column).

From the results shown in Table 4 it can be noted that, as in the case of the simulation experiments, all informed search methods outperform the baseline (UIS: uninformed/direct search). Interestingly, in the real-world experiments ISUP achieves a higher performance than AOSEOOR. ISUP performs well, but its detection rate is limited by the fact that it is reactive, i.e., it has no memory of the secondary objects seen previously. On the other hand, AOSEOOR uses the information on the secondary objects that have been observed, but it does not store a probability map. As a probability map for the main object is not available, and if no secondary objects are yet available as targets, the system follows a sub-optimal random path when searching for the main object.

The methods built using the methodology proposed here have a mixed performance, mainly because the use of negative information about the detection of the secondary object decreases the performance. The two methods that use just the positive information about the detection of the secondary object obtain the highest performance, ISCG-OP-SM being the best, with a 87% detection rate, meaning that the robot, when using this method, was able to detect the main object in 87% of the cases in less than 5 minutes. This performance is much higher than those of AOSEOOR and ISUP, which were 60% and 73%, respectively.

A very important observation is that the use of negative information increases the detection rate in the simulated experiments, but in the real world, it causes loss of performance. There are two main factors that explain this difference in performance. The first factor is that negative detections are very sensitive to some parameters in the model, like the maximum distance at which the object can be observed, and the probability of detecting the object when it is present. This fact can cause that perfect negative detection modeling in the simulator do not translate to a good model in the real world. Also, motion blur results in the maximum detection distance being variable and not a constant. The second factor is that the estimation of the pose of the robot is based on self-localization; so then inaccuracies in the pose of the robot can generate blurring in the probability map. And since the blurring tends to expand areas with low probability in the map, the use of negative detections of secondary objects, which have large detection cones, can diminish the result if they are further expanded. In consequence, the search algorithms work better in practice when negative information about secondary objects (that is hard to model) is not used. Also, this results in a faster algorithm, as convolutions need to be computed only when an object is detected.

In order to strength the previous results, an extra set of experiments (Set 2) was performed. The four methods that obtained the best results in the prior experiments (ISUP, AOSEEOR, ISC-OP-SM and ISG-O-SM) were selected, and 15 extra experiment trials were performed for each method. The results of the new experiments are shown in Table 5.

From the results shown in Table 5, it can be noted that ISCG-OP-SM is still the best method, obtaining an 87% detection rate. ISUP achieves a 53% detection rate, which is slightly better than ISC-OP-SM, which achieves a 47% detection rate. AOSEOOR is still the worst performing method, achieving a 27% detection rate. Although the detection rates suffered variations from Set 1 to Set 2, they are still comparable as the best and worst performing methods are preserved.

Table 4 Detection rates (%) for several methods for both simulation and for real world experiments

Method	Simulated Experiments with View Scale = 1.0	Real World Experiments (Set 1)
UIS	28	40
ISUP	31	73
AOSEOOR	51	60
ISC-PN-SM	53	47
ISC-OP-SM	42	80
ISCG-PN-SM	79	53
ISCG-OP-SM	74	87

Table 5 Detection rates (%) for the second set of real world experiments (Set 2)

Method	Real World Experiments (Set 2)
ISUP	53
AOSEOOR	27
ISC-OP-SM	47
ISCG-OP-SM	87

Then, from the analysis of both sets of experiments, the method ISCG-OP-SM is recommended when searching objects using the methodologies described in the present work. Note that both the simulations and the real experiments are complementary, as them shows both that the proposed methods improve over the preexistent ones, and that by improving the model of the perception process, negative information could become useful in real settings.

6 Conclusions

In this work, a Bayesian based methodology for performing informed object search was proposed. The methodology is based on integrating detections of secondary objects onto a main object probability map by using convolutions with spatial relation masks. A method for computing complex spatial relation masks by using a base composed of basic relation masks, and a database of co-occurrences of objects is used. Each basic relation mask corresponds to a spatial semantic category, such as ‘very near’, ‘near’, or ‘far’. In addition, a novel global search approach based on regions was proposed.

The proposed variants were compared both to a baseline (uninformed search), and with the systems of Aydemir et al. [8] (ISUP) and Elfring et al. [16] (AOSEOOR) in simulations and in the real world. Simulations allow performing replicable experiments and analyzing/simulating specific situations such as occlusions between objects, and misdetections that depend on the viewpoint of the simulated sensor with respect to the object. In simulations, the use of the proposed methodology increases the detection rate from 28% (uninformed search) to 79% (ISCG-PN-SM), when the main object can be detected at a maximum distance of 1 meter. This detection rate is also much higher than those obtained by ISUP (31%) and AOSEOOR (51%).

Real world experiments were performed by using a service robot in a domestic home-like environment. In the first set of real world experiments, the use of the proposed methodology increased the detection rate from 40% (uninformed search) to 87% (ISCG-OP-SM) when using convolutions, soft masks, global search, and information of only positive detection of secondary objects. The use of negative detections of secondary objects increased the detection rate in simulations but not in the real world, because detection model inaccuracies, motion blur and inaccuracies in localization change the behavior of the final probabilistic model. However, both in simulations and real experiments, the proposed methods consistently performed better than preexistent alternatives. A second set of experiments was performed for the best four algorithms. The detection rate obtained by ISCG-OP-SM is much higher than those obtained by ISUP and AOSEOOR in both sets of experiments, then it is the method recommended for performing search in the real world.

Future work for our system include performing more exhaustive real world experiments considering more objects, and using complementary features from the state of the art, like using object detectors based on convolutional neural networks, using voxels instead of grids, managing false detections, managing motion and map blur, using additional semantic categories/basic masks, improving the prior distribution by exploiting information about the environment, such as the presence of tables (they have a larger prior probability of containing an object than the floor), as well as modeling furniture as secondary objects. In this way, a table that is detected and is not present in the original map could be used as a secondary object, generating an increment in the probability distribution around it. Adaptation of spatial relations over time (e.g. by modifying their parameters adaptively) is another possible research topic. Also, as convolutions are performed on cells inside the viewpoint of the robot, all cells not yet observed by the robot have the same probability value. Then, the use of compressed representations like quadtrees/octrees could enable the robot to manage larger environments by using low-resolution representations on unvisited zones, and high-resolution representations on visited ones.

Acknowledgments This work was partially funded by Fondecyt Project 1130153 and Fondecyt Project 1161500.

References

1. Loncomilla, P., Saavedra, M., Ruiz-del-Solar, J.: Semantic object search using semantic categories and spatial relations between objects. In: RoboCup 2013: Robot World Cup XVII, pp. 516–527
2. Loncomilla, P., Saavedra, M., Ruiz del Solar, J.: A Bayesian framework for informed search using convolutions between observation likelihoods and spatial relation masks. In: 2013 16th International Conference on Advanced Robotics (ICAR), pp. 1–8 (2013)
3. Garvey, T.D.: Perceptual Strategies for Purposive Vision. Stanford University (1976)
4. Kasper, A., Jäkel, R., Dillmann, R.: Using spatial relations of objects in real world scenes for scene structuring and scene understanding. In: Proceedings of the 15th International Conference on Advanced Robotics, pp. 421–426 (2011)

5. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: KR'92 Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning, pp. 165–176 (1992)
6. Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: An overview. J. Fundamenta Informaticae - Qual. Spat. Reas. Archive **46**(1–2), 1–29 (2001)
7. Cohn, A.G., Li, S., Liu, W., Renz, J.: Reasoning about topological and cardinal direction relations between 2-dimensional spatial objects. J. Artif. Intell. Res. **51**, 493–532 (2014)
8. Aydemir, A., Sjöö, K., Jensfelt, P.: Object search on a mobile robot using relational spatial information. In: Proceedings 11th International Conference on Intelligent Autonomous Systems, pp. 111–120 (2010)
9. Aydemir, A., Sjöö, K., Folkesson, J., Pronobis, A., Jensfelt, P.: Search in the real world: Active visual object search based on spatial relations. In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 2818–2824 (2011)
10. Kunze, L., Doreswamy, K.K., Hawes, N.: Using qualitative spatial relations for indirect object search. In: Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 163–168 (2014)
11. Kunze, L., Burbridge, C., Hawes, N.: Bootstrapping probabilistic models of qualitative spatial relations for active visual object search. In: AAAI Spring Symposium, pp. 24–26
12. Li, J., Meger, D., Dudek, G.: Learning to generalize 3D spatial relationships. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 5744–5749 (2016)
13. Viswanathan, P., Meger, D., Southey, T., Little, J.J., Mackworth, A.: Automated spatialsemantic modeling with applications to place labeling and informed search. In: 2009 Canadian Conference on Computer and Robot Vision, pp. 284–291 (2009)
14. Russell, B., Torralba, A., Murphy, K., Freeman, W.: Labelme: A database and web-based tool for image annotation. Int. J. Comput. Vis. **77**, 157–173 (2008)
15. Zhou, K., Zillich, M., Zender, H., Vincze, M.: Web mining driven object locality knowledge acquisition for efficient robot behavior. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3962–3969 (2012)
16. Elfring, J., Jansen, S., van de Molengraft, R., Steinbuch, M.: Active object search exploiting probabilistic object-object relations. In: RoboCup 2013: Robot World Cup XVII, pp. 13–24 (2013)
17. Riazuelo, L., Tenorth, M., Di Marco, D., Salas, M., Gálvez-López, D., Mösenlechner, L., Kunze, L., Beetz, M., Tardós, J.D., Montano, L., Martínez Montiel, J.M.: RoboEarth semantic mapping: A cloud enabled knowledge-based approach. IEEE Trans. Autom. Sci. Eng. **12**(2) (2015)
18. Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernandez-Madrigal, J.A., Gonzalez, J.: Multi-hierarchical semantic maps for mobile robotics. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2278–2283 (2005)
19. Vasudevan, S., Gachter, S., Nguyen, V., Siegwart, R.: Cognitive maps for mobile robots-an object based approach. Robot. Auton. Syst. **55**, 359–371 (2007)
20. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**, 91–110 (2004)
21. Hanheide, M., Göbelbecker, M., Horn, G.S., Pronobis, A., Sjöö, K., Aydemir, A., Jensfelt, P., Gretton, C., Dearden, R., Janicek, M., Zender, H., Kruijff, G., Hawes, N., Wyatt, J.L.: Robot task planning and explanation in open and uncertain worlds. Artif. Intell. **247**, 119–150 (2015)
22. Wixson, L., Ballard, D.: Using intermediate object to improve efficiency of visual search. Int. J. Comput. Vis. 18 **3**, 209–230 (1994)
23. Ye, Y., Tsotsos, J.K.: Sensor planning for 3D object search. Comput. Vis. Image Underst. **73–2**, 145–168 (1999)
24. Shubina, K., Tsotsos, J.: Visual search for an object in a 3d environment using a mobile robot. Comput. Vis. Image Underst. 114 **5**, 535–547 (2010)
25. Aydemir, A., Pronobis, A., Gobelbecker, M., Jensfelt, P.: Active visual object search in unknown environments using uncertain semantics. IEEE Trans. Robot. **29**(4) (2013)
26. Ruiz-del-Solar, J., Loncomilla, P.: Robot head pose detection and gaze direction determination using local invariant features. Adv. Robot. **23**(2009), 305–328 (2009)
27. Martinez, L., Loncomilla, P., Ruiz del Solar, J.: Object recognition for manipulation tasks in real domestic settings: A comparative study. In: RoboCup 2014: Robot World Cup XVIII, pp. 207–219 (2015)
28. Loncomilla, P., Ruiz-del-Solar, J., Martinez, L.: Object recognition using local invariant features for robotic applications: A survey. Pattern Recog. **60**, 499–514 (2016)
29. Nash, A., Daniel, K., Koenig, S., Felner, A.: Theta*: Any-angle path planning on grids. In: AAAI'07 Proceedings of the 22nd National Conference on Artificial Intelligence, vol. 2, pp. 1177–1183 (2007)
30. Vaughan, R.T., Gerkey, B.P.: Reusable robot code and the player/stage project. In: Brugali, D. (ed.) Software Engineering for Experimental Robotics, ser. Springer Tracts on Advanced Robotics, pp. 267–289. Springer (2007)
31. Ludwig, O., Delgado, D., Goncalves, V., Nunes, U.: Trainable classifier-fusion schemes: an application to pedestrian detection. In: Proc. of the 12th Int. IEEE Conf. on Intell. Transportation Systems, vol. 1, pp. 432–437. St. Louis (2009)
32. Ruiz-del-Solar, J., Correa, M., Verschae, R., Bernuy, F., Loncomilla, P., Mascaró, M., Riquelme, R., Smith, F.: Bender – a general-purpose social robot with human-robot interaction abilities. J. Human – Robot Interact. **1**(2), 54–75 (2012)
33. Martinez, L., Matias P., Olave, G., Correa, M., Sanchez, L., Loncomilla, P., Ruiz-del-Solar, J.: UChile HomeBreakers 2015 Team Description Paper. Universidad de Chile, http://bender.li2.uchile.cl/Files/TDP/UChileHomeBreakers_TDP2015.pdf. Accessed 22 September 2017 (2015)

Patricio Loncomilla received the B.S., M.S., and Ph. D. degrees, all in electrical engineering, from the University of Chile, Santiago, Chile, in 2004, 2006, and 2011, respectively. Since 2011, he has been a researcher with the Advanced Mining Technology Center, University of Chile. His research interests include signal and image processing, pattern recognition, visual object recognition and representation, and simultaneous localization and mapping.

Javier Ruiz-del-Solar received his degree in Electrical Engineering from the Universidad Técnica Federico Santa María (Chile) in 1991, and the Doctor-Engineer degree from the Technical University of Berlin in 1997. Since 2009 he is Executive Director of the Advanced Mining Technology Center (AMTC) at the Universidad de Chile. His research interests include Mobile Robotics, Computer and Robot Vision, and Automation of Mining Equipment.

Marcelo Saavedra A. received his bachelor in science in System Engineering in La Paz, Bolivia in 2004. He received a graduate specialization degree in Robotics and computer vision from Alicante University in 2008. He also received his MSc. in Electrical Engineering, from the University of Chile, Santiago, Chile in 2015. He is currently professor at the Department of Mechatronics engineering of the Engineering Military School-EMI and La Salle University in La-Paz, Bolivia. His main research interests are in Modelling and Simulation, and Artificial Intelligence for Robotics.