Active Object Search Exploiting Probabilistic Object–Object Relations

Jos Elfring*, Simon Jansen, René van de Molengraft, and Maarten Steinbuch

Faculty of Mechanical Engineering, Eindhoven University of Technology,
5600 MB Eindhoven, The Netherlands
j.elfring@tue.nl

Abstract. This paper proposes a probabilistic object-object relation based approach for an active object search. An important role of mobile robots will be to perform object-related tasks and active object search strategies deal with the non-trivial task of finding an object in unstructured and dynamically changing environments. This work builds further upon an existing approach exploiting probabilistic object-room relations for selecting the room in which an object is expected to be. Learnt object-object relations allow to search for objects inside a room via a chain of intermediate objects. Simulations have been performed to investigate the effect of the camera quality on path length and failure rate. Furthermore, a comparison is made with a benchmark algorithm based the same prior knowledge but without using a chain of intermediate objects. An experiment shows the potential of the proposed approach on the AMIGO robot.

1 Introduction

Domestic robots are expected to operate in human populated environments. In order to successfully complete tasks, *e.g.*, delivering objects or safe navigation, accurate descriptions of such environments are required. However, due to the limited sensing range of robots and the fact that many of the objects involved get moved regularly, a complete description will never be available.

In this work the focus will be on active object search. More specifically, a robot will be given a task such as 'Find the book "Little Red Riding Hood"'. The robot then needs a strategy to find this object. A very naive approach would be to start an exhaustive search for the book through all the rooms. Even though the robot might very well succeed, this approach is not desired since many human users will have lost their patience long before the robot finishes its task. Two different situations can be distinguished. Firstly, the position of the target object can be known in advance. The search task then simplifies to a navigation task. Secondly, in the more challenging and realistic scenario, the location of the target object is not known in advance. In this second case humans have a high success rate and the search is efficient. Most probably this is due to

* Corresponding author.

S. Behnke et al. (Eds.): RoboCup 2013, LNAI 8371, pp. 13–24, 2014.

the fact that humans use experience, *e.g.*, books are often seen on a table or in the book case. In [17] it was shown both empirically and mathematically that exploiting indirect object relations increases performance. Representing human experience in a robot understandable way and exploiting it during search are some of the the main challenges in active object search.

If it comes to acquiring the common sense (or background) knowledge two possibilities are (i) learning or (ii) pre-programming it. The latter seems time consuming and will inevitably be incomplete. The first one however seems non-trivial since this knowledge is typically encoded in a human rather than robot understandable format. A popular alternative is to combine both. Various works [11,12] use human generated ontologies containing common sense knowledge, *e.g.*, 'coke is a drink' or 'a fridge is able to cool products'. In addition, this knowledge is combined with human generated knowledge available on the web. Among the popular alternatives are LabelMe [13], the Open Mind Common Sense (OMICS) [9] project and Flickr.com. In [15] LabelMe is used to get typical object-room relations, *e.g.*, 'kitchens typically contain a sink' or 'offices typically contain a bookshelf'. Afterwards, these statements are translated into probabilities. An approach leading to similar knowledge but using OMICS instead is presented in [11] whereas [10] uses Flickr for this purpose. Once this knowledge is represented by probabilistic models, the next step is exploiting it.

Already in the seventies, indirect object search was proposed as a way to limit the search space of the robot. In [3], the idea is to use an intermediate object that is easier to observe, *e.g.*, because it is larger, to find a target object. In case of the book, the robot could for example first try to find the, much larger, bookcase. This idea is the foundation of the strategy proposed in this work.

In [11], a decision-theoretic approach is presented for searching an object in large-scale indoor environments. Besides common sense knowledge, possibly complemented by robot observations, the robot also has a floor plan of the building available. Each room in the floor plan gets a probability of containing the target object based on the common sense knowledge. Both this probability and a utility function that incorporates the distance from the robot to the room are used to select the search location. An approach like this seems inevitable in large-scale environments, however, inside the rooms the robot still needs to perform an exhaustive search. Rather then selecting the most probable room, this work focuses on the search *inside* the most probable room.

In [4], a floor plan is not assumed to be available. After entering a room, it is classified autonomously and after classification a probability of containing the target object is calculated. Floor plans seldom change and for that reason the availability of a floor plan is considered reasonable. For that reason, this work aims at extending [11] rather than [4].

In [16,15] the focus is on place classification. Object-room relations are used for place classification using object detections. Similar knowledge is required and for that reason their approach for summarizing knowledge from LabelMe in probabilistic models is considered relevant in the context of this work. However, the problem investigated in this work is different and, for that reason, requires a different approach.

In [1] the focus is on organizing an efficient search given information about spatial relations between objects in an environment, *e.g.*, 'the book is on the table in room 1'. A decision theoretic strategy selection method is adopted for finding the object using the relations. The need of the spatial relations is considered to be too restrictive in the context of this work.

The application in [7,8] is finding target objects in a supermarket. Object-object attribute relations are used, rather then object-room relations, *e.g.*, 'an avocado will probably be located somewhere where other fruit can be found' or 'pizza can be found in a freezer'. A maximum entropy model steers the robot through the supermarket using these relations. The focus in this work is on the co-occurrence of objects rather than object-object attribute relations, since these are considered more useful in home or office like environments where objects are, contrary to supermarkets, not necessarily grouped by type.

Finally, [10] solves a similar problem with a different starting point, *i.e.*, a map containing objects is available. Based on a floor plan and the object locations, the best path is planned to a target object not included in the map based on the likelihoods of finding the object at each location in the map. This work differs from [10] by using a chain of intermediate objects and not assuming prior knowledge regarding object locations.

In order to find an object in large-scale indoor environments object-room relations are crucial for an efficient search. For that reason, this work builds upon the approach presented in [11]. However, [11] does not solve the problem of finding an object *inside* rooms. Indirect search using a single intermediate object is applied frequently but this work shows that it is even better to use a chain of intermediate objects, *e.g.*, books often lie at a nightstand, nightstands in turn are placed next to beds. Our first contribution is a strategy that allows using a chain of intermediate objects for active object search. A second contribution is a sensitivity analysis with respect to perception capability related parameters, *e.g.*, better camera or perception modules. Finally, we have performed an experiment to demonstrate the approach on a real robot.

The remainder of this paper is organized as follows. Section 2 explains how the object relations are learned and exploited during search. Section 3 explains the active object search strategy and Section 4 presents the findings obtained during an extensive performance analysis and comparison in simulation. Section 5 presents experimental results and Section 6 summarizes the conclusions and provides an outlook to future work.

2 Probabilistic Object Relations

2.1 Learning Object Relations

We are interested in two types of object relations. First of all, object-room relations and secondly, object-object relations. For the object-room relations we adopt the approach of [11]. In [11], human generated facts about typical combinations of rooms and objects stored in the OMICS database are matched to well-defined ontological concepts. Then the conditional probability of some object given a room is calculated by counting database entries and applying Lidstone's law with a λ of 0.5, according to Jeffrey-Perk's law, to compensate for unseen combinations.

To model object-object relations, the approach proposed in this work is to define the condition probabilities according to Lidstone's law [11,10]:

$$p(o_i \mid o_j) = \frac{N(o_i, o_j) + \lambda}{N(o_j) + \lambda n_{\text{obj}}}, \tag{1}$$

where both o_i and o_j are objects, $N(o_j)$ is the number of times object o_j is observed, $N(o_i, o_j)$ is the number of times objects o_i and o_j are observed in the same camera frame and n_{obj} is the number of objects. The dimensionless parameter λ can be used for smoothing. Furthermore, fading can be added to limit the sample size to grow unbounded [6], which is relevant in case of, *e.g.*, time-dependence. By counting the number of times objects are observed, visibility of objects is explicitly taken into account. Objects that are easily recognized get more relations and will for that reason be used as intermediate object more often.

In this work, more than 50.000 labeled images provided by LabelMe [13] are used to calculate the probabilities in (1). A robot can refine these *general* relations by updating the probabilities based on observed *instances* in its own environment and using (1). This way the general set of object–object relations changes according to the specifics of the robot's environment.

2.2 Exploiting Object Relations

In order to exploit the conditional probabilities defined in (1) for active object search, they must be reversed. The probability of finding the target object o_T if a set of objects $\mathcal{O}$ is observed, is of interest, *i.e.*, if the target object is observed, what is the probability of observing the object currently in sight. Using Bayes' law:

$$p(o_j \mid o_T) = \frac{p(o_T \mid o_j)\, p(o_j)}{\sum_j p(o_T \mid o_j)\, p(o_j)}. \tag{2}$$

$\mathcal{O}$ is a set containing all objects observed by the robot and the summation over j represents the summation over all members of $\mathcal{O}$.

For selecting the room to which to navigate, the decision-theoretic approach introduced in [11] is adopted. For determining to which object to navigate inside that room, an expected utility, inspired by [11,7], that combines the probability of successfully finding the target object with the travel cost is maximized:

$$o^* = \operatorname*{argmax}_{o_i \in \mathcal{O}} \left[p(o_i \mid o_T) + w \cdot V(o_i) \right], \tag{3}$$

where again o_T is the object that has to be found, o_i is an object in the set $\mathcal{O}$, w acts both as a relative weight and a scaling term and the travel cost is defined as the reciprocal of the arc tangent of the Euclidian distance $d_{r,o}$ from robot to object:

$$V(o) = \frac{1}{\arctan(d_{r,o})}. \tag{4}$$

The weighted sum in (3) balances between distance and probability and avoids oscillating behavior. The expected utility for some object o is represented by $\hat{U}(o)$ and given by the expression between squared brackets in (3).

3 Active Object Search Strategy

Section 2.2 introduced (3) for calculating which object from the set of observed objects $\mathcal{O}$ maximizes the expected utility. The next step is incorporating this in an active object search strategy. The first step of this strategy is to determine the room in which the object is expected to be according to the approach of [11]. As before, $\mathcal{O}$ is the set containing the objects observed by the robot. $\mathcal{F}$ is the set containing objects marked as dead end, initially $\mathcal{F} = \emptyset$. $\mathcal{T}$ is the set containing the (intermediate) target object(s), initially $\mathcal{T} = \{o_T\}$. If $|\mathcal{T}| > 1$, all members are considered in (3). The proposed search strategy is as follows:

1. $\mathcal{O} \leftarrow \mathcal{O} \backslash \mathcal{F}$, then determine o^* using (3).

2a. If $\hat{U}(o^*) \geq$ *threshold*, start navigating towards object o^*. If during navigation new objects appear, add them to $\mathcal{O}$ and go to step 1 to see if the navigation goal has to be updated.

2b. If $\hat{U}(o^*) <$ *threshold*, the observed objects have no sufficient relation with the target object and an intermediate object needs to be added. The object o_s with strongest relation to any of the objects in $\mathcal{T}$ according to Bayes' law is added to $\mathcal{T}$: $\mathcal{T} \leftarrow \mathcal{T} \cup \{o_s\}$. With $\mathcal{T}$ updated, go to step 1.

3a. If the robot reaches an intermediate object it is added to $\mathcal{F}$ since it did not lead to the target object. Now go back to step 1 and try to find a route via another object. If such route is not available, start a random search and update $\mathcal{O}$ if new objects are found.

3b. If the robot during navigation finds the target object, the task is succeeded

4. If the search fails, try the next best room.

For an illustrative example consider Figure 1. Both $p(o_T \mid A)$ and $p(o_T \mid D)$ are defined 0.45, $p(o_T \mid B) = 0.1$ and $p(o_T \mid C) = 0$. In Figure 1(a) the robot observes objects A, B and C. The robot moves via B to A because of the travel cost in (3). In Figure 1(b) object B is set as a dead end. After falsifying object A, the robot only observes C, however, objects C and o_T do not have a direct link. As a result, the most probable neighbor of the target object, *i.e.*, object D since A is a dead end already, is added to the target set. This object has a relation with object C, hence the robot moves towards C, see Figure 1(c). While doing so, object D gets observed and the robot starts moving towards D. Before arriving at D, the target object is observed and the search is succeeded. This example shows how a target object can be found using a chain of intermediate objects.

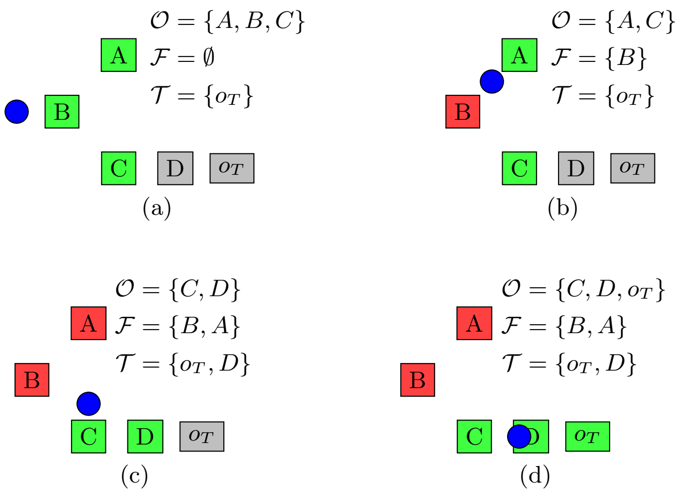


Fig. 1. Illustrative example of the search strategy. The robot is represented by the blue dot, green objects are in sight, red objects are dead ends. The target object o_T is assumed to have a 0.45 change being either next to D or A. Via a chain of intermediate objects, the target object is found.

4 Simulation Results

A large number of simulations is performed to (i) investigate the effect of the robot's sensors on the success of the search and (ii) compare the proposed strategy with a benchmark strategy. Simulations enable a high number of trials under the exact same conditions, thereby allowing for a fair comparison. Section 4.1 explains how we have updated the general relations using simulated data. After that, Section 4.2 briefly explains a benchmark search strategy and Section 4.3 presents the simulation results.

4.1 Simulation Set-Up

For all simulations, the general relations from LabelMe are refined by adding a large number of observations in different but similar simulated rooms. One simulated room, referred to as recipe room, was created. Based on a typical living room ten main objects were placed inside this room, *e.g.*, a dining table, two couches, etc. Furthermore twenty smaller objects were defined with locations relative to one or more main objects, *e.g.*, chairs stand next to the dining table, keys can be on any of four tables. All positions are disturbed by Gaussian noise. The recipe room was used to create 30 random rooms in which the object–object relations were learned. The simulated robot has a 360° viewing angle and the target object was always present in the room. Table 1 shows some of the typical condition probabilities of observing two objects in the same camera frame learned from the recipe room. A typical path towards the keys resulting from the proposed strategy is, *e.g.*, from the dining table towards the three seater via the chair, then from three seater to the coffee table on which the keys are found.

Table 1. Example probabilities learned from simulation

	Coffee table	Side table	Two seater
Book	0.20	0.33	0.11
Three seater	0.23	0.11	0.12
Keys	0.21	0.31	0.09
Remote control	0.01	0.19	0.13

For validation purposes, a second data set was obtained by redesigning the recipe room, *i.e.*, the dining table moved to the other side of the room and couches in a different formation. As a result, the smaller objects changed position too.

In order to be able to analyze the effect of the robot's sensors the view scale is varied. The view scale VS, in [14] referred to as effective range of detection, can be interpreted as the simulated resolution of the camera and is calculated as:

$$VS = \frac{d_{\text{max}}}{S_{\text{avg}}}, \tag{5}$$

where d_{max} is the maximum distance from which an object with average size S_{avg} is detectable. VS is assumed independent of the object class and will be varied during the simulations. In [14] VS lies around 12, a Microsoft Kinect with object recognition using SIFT feature points has VS around 5 for typical household items on our robot.

4.2 Benchmark Search Strategy

For comparison a search strategy only exploiting direct neighbors was implemented, *i.e.*, a strategy that does not allow for a chain of intermediate objects. The direct neighbors in both strategies are the same. The strategy is as follows:

1. Check the probabilities of the objects neighboring the target object
2. Check the camera image for objects

3a. If none of the neighboring objects is in field of view, move to the center of the room while looking for objects

3b. If at least one neighboring object is found, navigate towards the most probable neighboring object

4a. If the robot during navigation finds the target object, the task is succeeded

4b. If the neighboring object is reached without observing the target, try another neighboring object

4c. If no new neighboring objects are found, start a random search

4.3 Results

Both the benchmark strategy and the strategy introduced in this work are tested on failure rate. The failure rate is the percentage of search tasks that would eventually end up with a random object search. The camera *VS* during the learning of the relations was arbitrarily set to 4. The effect of *VS* during learning will be investigated later. After that, the robot was asked to find the remote control, which has various possible locations as was already shown in Table 1. To investigate the effect of the *VS* the failure rate was calculated for 27 different values of *VS* in 25 different rooms per *VS*. Each room was entered from each corner for both methods, hence the results are an average of 5400 simulations. The results are shown in Figure 2.

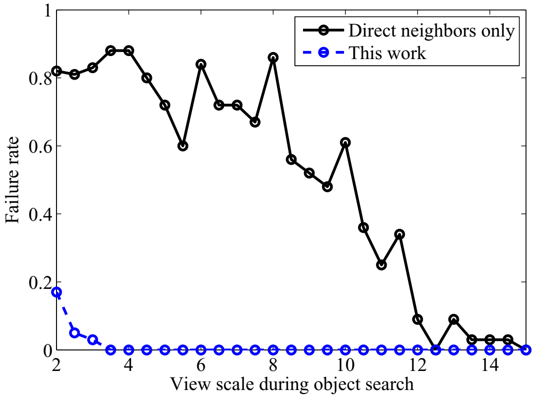


Fig. 2. Failure rate as a function of *VS* during the active object search

This figure shows a failure rate that rapidly goes to zero for the proposed approach. Only for very low values of *VS*, *i.e.*, a very limited number of object detections due to the very limited sensing range, the search failed. For more realistic values of *VS* the robot was always able to find a chain of intermediate objects leading to the target object, which means that the robot was performing a directed search. For the benchmark strategy only exploiting direct relations the failure rate is much worse. This is due to the fact that direct neighbors of the remote control, being the coffee table, side table, two seater and tv, are only observable in a relatively small portion of the room. The noisy line is caused by the sensitivity to the configuration of the limited number of direct neighbors. As *VS* increases, objects can be detected from larger distances and the failure rate drops. Increasing the number of 25 different rooms per configuration will lower the noise, increasing the room size will enlarge the differences.

Figure 3 shows the path lengths for the same set of simulations ignoring those that failed. For comparison, the path length using a very naive zigzag search strategy was added.

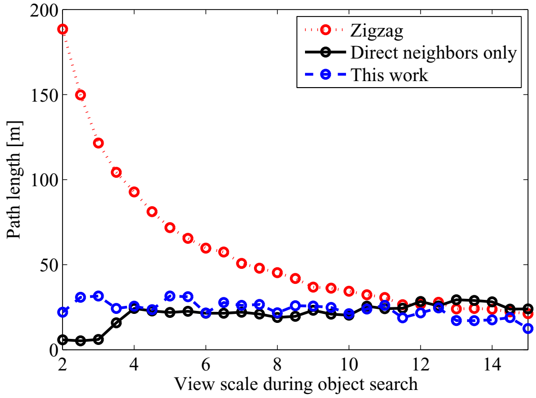


Fig. 3. Path length as a function of VS during the active object search

For low values of VS the benchmark approach seems to outperform the others, however, this method fails most of the time. The trails that succeeded started from a corner which appeared to be very close to the target object and as a result the average path length during the successful trials is very low. In case of the numerous failures, a random search has to be started and that will require many additional meters as can be seen from the red line. Such random search is however not yet combined with the benchmark strategy to allow for a fair comparison. The benchmark strategy does not incorporate the distance from robot to an object and for that reason, it performs worse than the naive zigzag strategy for very high values of VS. The proposed approach appears to be insensitive for the performance of the camera and is likely to outperform the benchmark approach over the full range of possible VS values. For very high values of VS, objects can be seen from many places in the room and for that reason, the differences are getting smaller.

To further investigate the effect of VS on the performance, the experiment above was repeated with relations that were learned by a robot with $VS = [5, 10, 15]$. The object search is again performed for 27 values of VS and starting from each corner of the room. Again each setting was simulated 25 times hence the average results of 8100 simulations are shown in Figure 4.

Two conclusions can be drawn from this figure. First of all the value of VS during learning does not have an effect on the path length during the active object search. As a result, it should be possible to share the relations among different robots with different perceptual capabilities without negatively influencing the performance. Secondly, the total path length decreases for VS above 9 or 10. Due to the different scale on the vertical axis, this conclusion can hardly be made from Figure 3 that partly shows the same information.

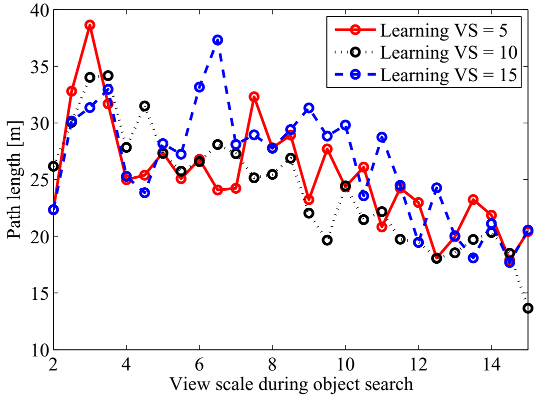


Fig. 4. Effect of VS during learning on the path length as a function of VS during the active object search

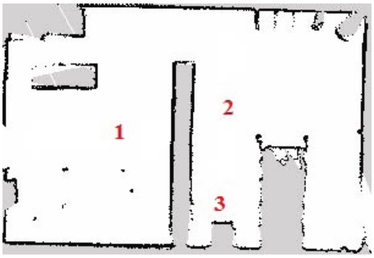
5 Experiment

In order to test the active search strategy on a real robot, our AMIGO robot was asked to find a lamp. The relations used in this experiment are learned from the LabelMe data set. Existing LabelMe tools [13] are used to unify the annotations, *e.g.*, nightstand and bedstand are synonyms. Relations are only considered if $N(o_i, o_j) \geq 5$ to avoid atypical object–object relations that might appear in the LabelMe database. The experiment is carried out in an atypical lab environment and for that reason, the room selection strategy is omitted in this experiment and the number of objects is somewhat lower than in a real world environment.

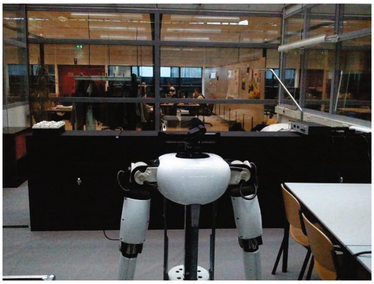
All perception on the robot uses RGB-D data obtained via a Kinect. By detecting horizontal planes at heights in a certain range, tables are detected. Objects are recognized using color information and/or using both 2D and 3D templates [5], anchoring, tracking and data association is performed using [2].

Figure 5 shows both a photograph of and a occupancy grid map of the lab. The numbers in the list below refer to the positions in the lab indicated in Figure 5(a):

1. Inside the lab, AMIGO detects a table and the bed. The table has a direct relation with the target object and for that reason, AMIGO navigates towards the table. On top of the table, no lamp was observed. Bed has no relations with lamp and for that reason, the most probable neighbors of lamp, being table and ceiling, are added as intermediate object. A ceiling is not observed, whereas the bed is, hence AMIGO navigates towards the bed.
2. While navigating towards the bed, the small table next to the bed is observed.
3. While navigating towards this table, the lamp is found on top of it.



(a) Occupancy grid map of the lab in which the search takes place.



(b) Photograph of the lab with AMIGO from the entrance.

Fig. 5. Lab environment in which the experiment is carried out

6 Conclusion and Future Work

This work introduced an active object search strategy exploiting probabilistic object-object relations learned based on their co-occurrence in camera images. The strategy is particularly useful inside rooms and is combined with the strategy introduced in [11], which calculates in which room to expect an object. The strategy is extensively analyzed in a large number of simulations. Simulations showed (1) the advantage of using a chain of intermediate objects over using direct neighbors only, (2) better perceptual capabilities lead to shorter paths during the search and (3) the perceptual capabilities of a robot during the learning of the relations do not influence the performance during the search. An experiment involving a real robot has shown the potential of the approach in the real world.

Future work can be to further validate the performance on real robots. Furthermore, the dependency of object–object relations on the room type should be investigated.

Acknowledgment. The research leading to these results has received funding from the European Union Seventh Framework Program FP7/2007-2013 under grant agreement no. 248942 RoboEarth.

References

1. Aydemir, A., Sjöö, K., Folkesson, J., Pronobis, A., Jensfelt, P.: Search in the real world: Active visual object search based on spatial relations. In: 2011 IEEE International Conference on Robotics and Automation, pp. 2818–2824 (2011)
2. Elfring, J., van den Dries, S., van de Molengraft, M., Steinbuch, M.: Semantic world modeling using probabilistic multiple hypothesis anchoring. Robotics and Autonomous Systems 61(2), 95–105 (2013)
3. Garvey, T.D.: Perceptual strategies for purposive vision. Ph.D. thesis, Stanford University, Stanford, CA, USA, aAI7613006 (1976)

4. Hanheide, M., Gretton, C., Dearden, R.W., Hawes, N.A., Wyatt, J.L., Pronobis, A., Aydemir, A., Göbelbecker, M., Zender, H.: Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), Barcelona, Spain (July 2011)
5. Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., Lepetit, V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 858–865 (November 2011)
6. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs. Springer (2007)
7. Joho, D., Burgard, W.: Searching for objects: Combining multiple cues to object locations using a maximum entropy model. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 723–728 (May 2010)
8. Joho, D., Senk, M., Burgard, W.: Learning search heuristics for finding objects in structured environments. Robotics and Autonomous Systems 59(5), 319–328 (2011); special Issue ECMR 2009
9. Kochenderfer, M.J., Gupta, R.: Common sense data acquisition for indoor mobile robots. In: Nineteenth National Conference on Artificial Intelligence, AAAI 2004, pp. 605–610. AAAI Press / The MIT Press (2003)
10. Kollar, T., Roy, N.: Utilizing object-object and object-scene context when planning to find things. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 2168–2173 (May 2009)
11. Kunze, L., Beetz, M., Saito, M., Azuma, H., Okada, K., Inaba, M.: Searching objects in large-scale indoor environments: A decision-theoretic approach. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 4385–4390 (May 2012)
12. Pronobis, A., Jensfelt, P.: Large-scale semantic mapping and reasoning with heterogeneous modalities. In: Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA 2012), Saint Paul, MN, USA, pp. 3515–3522 (May 2012)
13. Russell, B., Torralba, A., Murphy, K., Freeman, W.: Labelme: A database and web-based tool for image annotation. International Journal of Computer Vision 77, 157–173 (2008)
14. Tsotsos, J.K., Shubina, K.: Attention and Visual Search: Active Robotic Vision Systems that Search. In: 5th International Conference on Computer Vision Systems Conference (March 2007), `http://biecoll.ub.uni-bielefeld.de/volltexte/2007/1`
15. Viswanathan, P., Southey, T., Little, J., Mackworth, A.: Place classification using visual object categorization and global information. In: 2011 Canadian Conference on Computer and Robot Vision (CRV), pp. 1–7 (May 2011)
16. Viswanathan, P., Southey, T., Little, J., Mackworth, A.: Automated place classification using object detection. In: 2010 Canadian Conference on Computer and Robot Vision (CRV), May 31-June 2, vol. 2, pp. 324–330 (2010)
17. Wixson, L.E., Ballard, D.H.: Using intermediate objects to improve the efficiency of visual search. International Journal of Computer Vision 12, 209–230 (1994)