Indirect Object Search based on Qualitative Spatial Relations

Lars Kunze[⋆] and Nick Hawes[⋆]

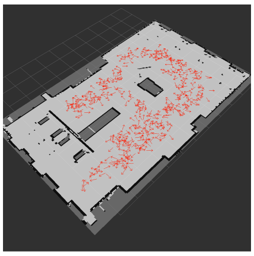
***Abstract*— Accomplishing object search tasks in human environments requires autonomous mobile robots to reason about potential object locations and to plan for the next best view accordingly. By using information about the 3D structure of the environment, knowledge about landmark objects and their spatial relationship to the sought object, the search can be improved by directing the robot towards the most likely object locations.**

In this paper we have designed, implemented and evaluated an approach for searching for objects on the basis of Qualitative Spatial Relations (QSRs) such as *left-of* and *close-to*. On the basis of QSRs between a landmark and the sought object we generate a Gaussian Mixture Model (GMM) for representing metric poses of potential object locations using a ternary point calculus. The GMM is employed within object search for planning the next best view. Preliminary results show that search methods based on QSRs are faster and more reliably than methods not considering them.

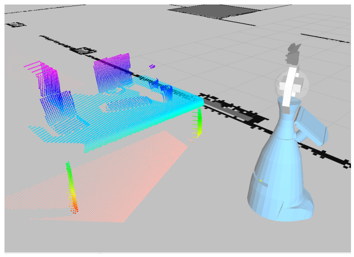
I. INTRODUCTION

In recent years, we have seen substantial progress towards personal robot assistants performing everyday tasks in human environments. The ability to *search for objects* is an integral part of many of those tasks, as the locations of task-relevant objects are often unknown beforehand, or change over time. Although the locations of some objects often change due to the dynamics in human environments (e.g. pens, mugs, books, magazines etc.), the locations of other objects are more consistent (TVs, desks, cabinets etc.). Let us, for example, consider an office environment which contains both largely stationary items (furniture such as office desks, cupboards and drawers, and devices such as desktop PCs, monitors and printers) and movable table-top objects (such as pencils, papers and cups). If a robot has some knowledge about the relationships between such objects in the environment, it can exploit this information when searching for an object. For example, a robot supposed to search and locate unused coffee cups in an office space could make use of the information that coffee cups are often located on offices desks, to the left or to the right of a keyboard and in front of a monitor. Given this knowledge about the environment, the robot can identify potential object locations and reason about its own goal locations in order to perceive the object under consideration. Narrowing the target of an object search through the use of an intermediate object is known as *indirect search*, and has been previously shown to be an effective method of improving search performance [1].

In this paper we investigate how indirect search using *Qualitative Spatial Relations* (QSRs [2]) can improve the performance of a robot searching for objects in human environments. To this end, we have designed, implemented and evaluated an approach for searching for objects based on QSRs. Figure 1 illustrates the questions a robot has to answer when searching for an object, namely where to stand and where to look. In this work we are answering these questions on the basis of QSRs. Extending the approach of Aydemir et al. [3], we assume that the robot has the following information at the beginning of the search: a 2D map, a 3D voxel-based occupancy map, a set of poses of known landmark objects, and a set of task-relevant QSRs. Given this information, the general idea of QSR-based object search is as follows. First, the robot samples a set of random poses within the free space in the 2D map. It then evaluates each sampled pose based on the view it provides for object search with respect to the 3D map and the QSRs. Then the robot selects the best view, navigates to it, and uses its perception routines to attempt to recognize the sought object. This procedure is repeated until the robot either finds the sought object or the process is aborted.



(a) **Q:** Where to stand to find a keyboard? **A:** Keyboards are usually *on* supporting surfaces such as tables.



(b) **Q:** Where to look for it? **A:** Keyboards are likely to be found in front of a monitor.

Fig. 1. Searching objects in human environments requires autonomous mobile robots to reason and to plan where to stand and where to look. This paper investigates how the search for objects can be improved by using 3D environments maps, information about landmarks and qualitative spatial relations between objects.

The remainder of the paper is structured as follows. In Section II, we present the underlying representations of QSRs, explain how we generate potential object locations based on a ternary point calculus, and show how we use the generated information within the search for objects. In Section III, we present preliminary results of different search methods employed in simulated object search experiments. In Section IV, we put this work into context by discussing related approaches before we conclude in Section V.

[⋆]Intelligent Robotics Lab, School of Computer Science, University of Birmingham, United Kingdom {l.kunze|n.a.hawes}@cs.bham.ac.uk

Fig. 2. Different geometric configurations of object on an office desk on the basis of the same QSRs.

II. QSR-BASED INDIRECT OBJECT SEARCH

As stated previously, the idea of QSR-based indirect object search is as follows: the robot searches for an object by considering the QSRs between the sought object and other objects in the environment that function as landmarks. In this work, we distinguish between two different object classes, namely, *static* and *dynamic* objects. In Table I we give some examples of different types of objects. We refer to objects as *static* if they are only subject to quantitative location changes, but not subject to qualitative changes. On the other hand, we refer to objects as *dynamic* if they are subject to qualitative location changes. In our approach we use *static* objects as landmarks for finding the locations of *dynamic* objects on the basis of QSRs.

TABLE I
LOCATION CHANGES OF OBJECT TYPES

Object type	Location changes	Examples
static	quantitative	desktop PC, monitor, printer
dynamic	qualitative	keyboard, mouse, cup,

In [4], the authors use topological relations such as "in" and "on" to specify potential object locations. In this work we use directional (*left-of, right-of, in-front-of, behind-of*) and distance relations (*close-to, distant-from*) to describe the QSRs between objects. Figure 2 shows different geometrical configurations of a desktop scene with the same underlying QSRs. In this example, the monitor functions as primary landmark object and the scene is defined by the following QSRs:

$$\begin{aligned}&\mathit{scene}(\mathit{Monitor}, \mathit{Keyboard}, \mathit{Laptop}, \mathit{Cup}, \mathit{Bottle}) \Leftrightarrow\\ &\quad \mathit{in\text{-}front\text{-}of}(\mathit{Keyboard}, \mathit{Monitor}) \wedge\\ &\quad \mathit{left\text{-}of}(\mathit{Laptop}, \mathit{Keyboard}) \wedge\\ &\quad \mathit{right\text{-}of}(\mathit{Cup}, \mathit{Keyboard}) \wedge\\ &\quad \mathit{behind\text{-}of}(\mathit{Bottle}, \mathit{Cup}) \wedge\\ &\quad \mathit{close\text{-}to}(\mathit{Bottle}, \mathit{Cup}).\end{aligned}$$

In order to utilize this information within object search tasks, the symbolic scene description from above is transformed into a sub-symbolic representation that can directly be integrated with the environment model of the robot.

To this end, we use the qualitative positional calculus based on ternary relations [5] that has been developed in the context of robot navigation. The three positions in the calculus are referred by *origin*, *relatum* and *referent*. The *origin* corresponds to the position of the robot. *Origin* and *relatum* define the reference axis which partitions the surrounding space. Then, the spatial relation is defined by the partition in which *referent* lies with respect to the reference axis. In order to determine the partition, i.e. the directional relation, [5] calculate the relative angle ϕ_{rel} as follows:

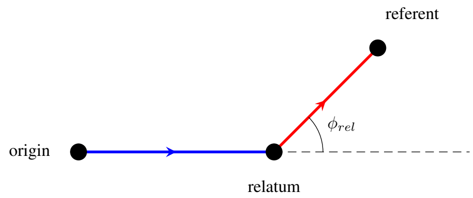


Fig. 3. The relative angle ϕ_{rel} is defined by the reference axis, which is specified by *origin* and *relatum*, and the *referent*. The example above illustrates a situation where the *referent* is left and behind of the *relatum*.

$$\phi_{rel} = \tan^{-1}\frac{y_{ref} - y_{rel}}{x_{ref} - x_{rel}} - \tan^{-1}\frac{y_{rel} - y_{orig}}{x_{rel} - x_{orig}} \quad (1)$$

ϕ_{rel}, is the angle between the reference axis, defined by *origin* and *relatum*, and the *referent* point. This is visualized in Figure 3.

In our work, we assume that the robot position (*origin*), is located in front of the office desk, facing the intrinsic front side of the monitor (*relatum*). Given this idealized situation, the robot hallucinates potential object locations of, for example, a cup on the basis of QSRs. That is, the robot first generates the potential locations of the keyboard, and afterwards a potential location of a cup.

We generate geometric positions from QSRs by sampling the relative angle ϕ_{rel} from a set of Gaussian distributions representing the directional relations. The four directional relations *behind-of*, *left-of*, *in-front-of*, and *right-of* are represented by Gaussian normal distributions with means 0, $\frac{1}{2}\pi$, π, and $\frac{3}{2}\pi$ respectively. For generating the distances for the proximal relations between objects we used a similar approach while taking the dimensions of a hypothetical office desk into account. We determine an axis between the landmark and the edge of the desk in the direction of the relative angle ϕ_{rel}. This axis is divided into two qualitatively different intervals: *close-to* and *distant-from*. For each interval we use a uniform distribution to sample a distance between the object and the landmark according to the QSRs.

Having sampled a number of potential object positions we represent them by a single multivariate Gaussian distribution relative to the landmark. Different QSRs are represented by combining the individual Gaussians into a Gaussian Mixture Model (GMM):

$$P_{QSR_{rel}}(\mathbf{x}|\lambda) = \sum_{i=1}^{m} w_i \mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i), \quad (2)$$

where $\mathbf{x}$ denotes the relative object position with respect to the landmark and λ is a set of parameters $\{w_i, \mu_i, \Sigma_i\}$ for m Gaussian distributions. The weight w_i of each Gaussian is determined by dividing the number of samples for a

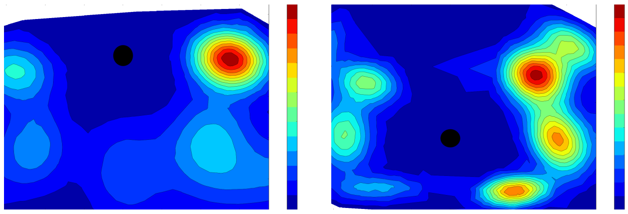


Fig. 4. Gaussian Mixture Models (GMMs) generated from QSRs. The images depict a top-down view onto a table whereby the image width corresponds to 1.5 meters. The landmarks are indicated by a black circle. Left: Locations of a cup with respect to a monitor. Right: Locations of a cup with respect to a keyboard.

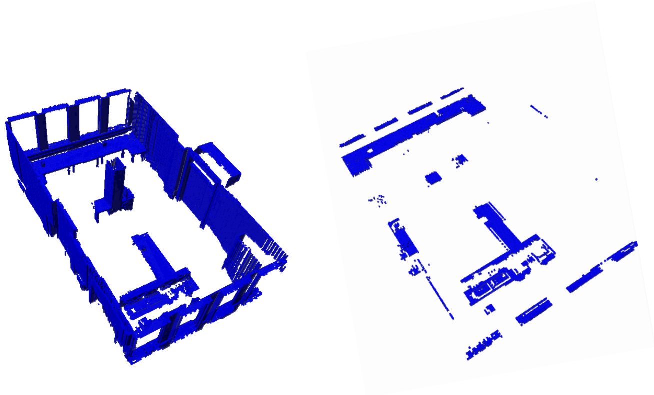


Fig. 5. Left: 3D Octomap of the environment. Right: Extracted supporting planes on the basis of a normal estimation.

particular QSR by the total number of samples. Thereby the weights w_i (for $i = 1, \dots, m$) always sum up to one. For example, if we sample the QSR *left-of(Cup,Keyboard)* 15 times, and the relation *right-of(Cup,Keyboard)* 85 times, then the related Gaussians are weighted by 0.15 and 0.85 respectively. Figure 4 visualizes GMMs for a cup with respect to two landmarks: monitor (left) and keyboard (right).

The overall set of QSRs is then represented by a mixture of 2D Gaussians. How this model is used within a search task is explained in the next section.

A. Search Method

Before we outline the view planning algorithm in detail we recapitulate the different types of information the robot uses for reasoning about the potential object locations: a 2D map, a 3D occupancy map, a set of landmark objects, and a mixture of Gaussians generated from the QSRs.

When the search is started, the robot first receives the latest version of the 3D occupancy map, calculates the average normal for each voxel in the map, and keeps only those voxels which normals are pointing upwards (in a certain range). These voxels are considered as part of supporting planes. We denote these voxels by $v_1 \dots v_k$. Figure 5 shows the complete 3D Octomap [6] of the environment (left) and the extracted supporting planes according to the averaged normals (right).

In a second step, the robot samples n navigatable poses from the 2D map which are denoted by Ψ. At each of

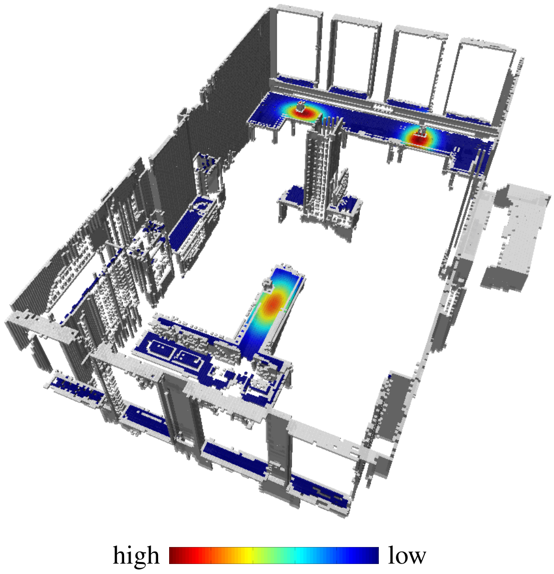


Fig. 6. Voxels of the supporting planes are weighted by the Gaussian Mixture Model (GMM) derived from the QSRs

these poses, we calculate a 2D view cone according to the robot's sensor specification. The view cones are evaluated with respect to a 2D projection of the 3D occupancy map. To assess the view cones, we count the number of occupied voxels that lie within a cone. However, we only consider voxels that have been classified as part of a supporting plane beforehand. Furthermore, the voxels are weighted in regards to the QSR (GMM) models. The function $Viewcone(\psi)$ returns the view cone of the robot at a given pose ψ. And the function $In(v_i, Viewcone(\psi))$ returns 1, if the voxel v_i is in the view cone of pose ψ, otherwise 0. To select the best view cone the robot uses the equation as defined below:

$$\underset{\psi \in \Psi}{\operatorname{argmax}} \sum P_{QSR}(v_i) In(v_i, Viewcone(\psi)) \qquad (3)$$

where $P_{QSR}(v_i)$ denotes a probability distribution in the world frame to find the object at voxel v_i. This distribution is calculated by placing the the relative QSR-based mixture models ($P_{QSR_{rel}}$) at the poses of the known landmarks. Figure 6 visualizes an example of such a probability distribution over the voxels that had been classified as supporting planes on the basis of the mixture of Gaussians generated from the QSRs. Having selected a best view cone the robot proceeds by navigating to the respective pose and by running its perception routines. The overall algorithm of the view planning procedure is formalized in Algorithm 1.

Figure 7 and Figure 8 visualize the progress of a search and the evaluated view cones with respect to the supporting planes and the QSRs respectively. The colors of the view cones indicate the probability to find an object at the respective poses. When comparing the highly rated view cones from both figures it is visible that the QSR-based view cones are much more directed towards the Gaussian Mixture Model shown in Figure 6.

Algorithm 1 QSR-based object search

Require: M_{2D} and M_{3D} are the 2D and 3D environment maps respectively; n denotes the number of poses to be sampled

procedure SELECTBESTVIEW(n, M_{2D}, M_{3D}, P_{QSR})
  $\{v_1 \dots v_k\} \leftarrow VoxelsOfSupportingPlanes(M_{3D})$
  $\Psi \leftarrow NavigatablePoses(n, M_{2D})$
  Initialize $sum(\psi)$ with 0 for all poses $\psi \in \Psi$
  for all $\psi \in \Psi$ **do**
    $vc \leftarrow Viewcone(\psi)$
    for all $v_i \in \{v_1 \dots v_k\}$ **do**
      $sum(\psi) \leftarrow sum(\psi) + P_{QSR}(v_i) \times In(v_i, vc)$
    end for
  end for
  Find $\psi^\star$, that maximizes $sum(\psi)$ for all $\psi \in \Psi$
  return $\psi^\star$
end procedure

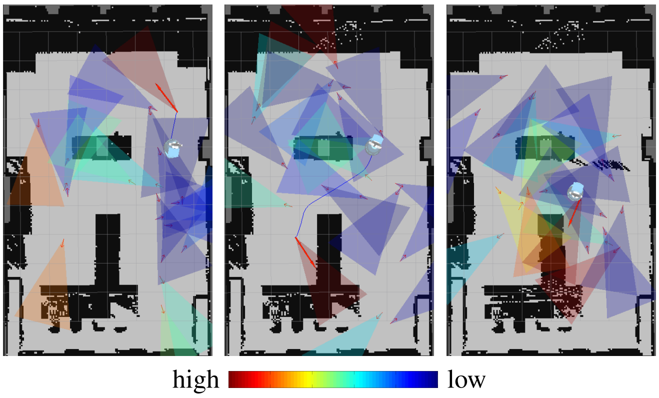


Fig. 7. Viewcone evaluation on the basis of a uniform distribution with respect to supporting planes. Search over three poses.

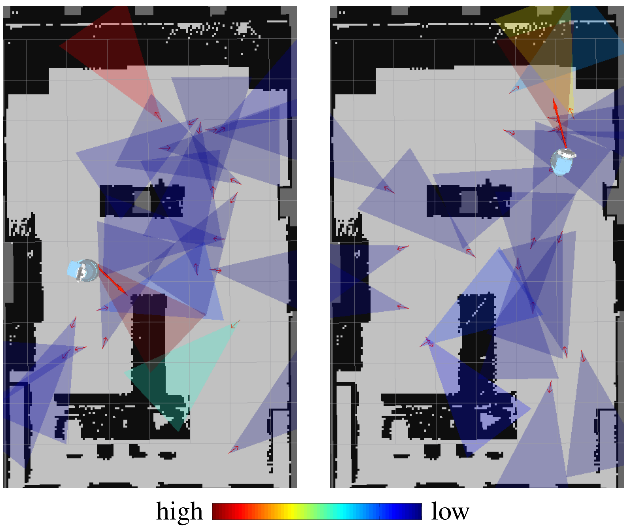


Fig. 8. View cone evaluation on the basis of QSR-based models.

III. PRELIMINARY RESULTS

The QSR-based search method described in the previous section has been implemented and evaluated in a simulated environment. We used the open source robot simulator MORSE [7] for simulating the environment[1], the SCITOS G5 robot platform[2] and its sensors. In simulation, we used a semantic camera to perceive objects in the environment. The semantic camera returns an object ID, the object's type, and its pose whenever an object is in sight and between the near and far plane of the camera's view frustum.

In the experiments, the robot was controlled through the task-level architecture SMACH[3] and the middleware ROS[4]. The robot control program is comprised of four states: a search monitor, a particular search method, a navigation routine, and a perception routine. The search monitor assesses the overall progress of the search, i.e., whether an object was found or not and/or whether a timeout has occurred. On this basis it decides to continue or to abort the search task. If it decides to continue the search, the search method selects the next best view pose and the navigation routine moves the robot to the goal accordingly. At the goal location the perception routine is called and the result is interpreted by the search monitor and so on.

In our experiment we compared three different search methods: a purely random method, a method based on the information about supporting planes and the QSR-based method described in the previous section:

- Within the *random* search method 20 locations are sampled from the 2D map. Out of the 20, a single goal location is randomly selected and sent to the robot's navigation routine.
- Within the *supporting planes* method 20 locations are sampled from the 2D map and evaluated with respect to the projected 3D occupancy map of voxels that had been classified as supporting planes.
- Within the QSRs-*based* method also 20 locations are sampled from the 2D map and evaluated with respect to the 3D voxel weighted according to the QSR-based mixture of Gaussians.

Table II summarizes the results of ten searches using each search method respectively. In total we placed three cups in the environment. If the object was not found within a time span of two minutes the search was aborted. First it can be noted that the uninformed random search method, namely *random*, was only able to find the object in 60% of the searches, that is, 40% of the searches had been aborted. Please note, that the average time and the average number of searched poses was calculated on the basis of successful trials only. The average time also includes the time for making the QSR-based inferences based on the pre-defined models. The second noteworthy aspect of the results is how the average time and average number of searched poses decreased when

[1]In the experiments we used the TUM kitchen environment of MORSE
[2]`http://metralabs.com/`
[3]`http://wiki.ros.org/smach`
[4]`http://wiki.ros.org/`

more information is considered in the view planning step. Although both informed search methods, namely *supporting planes* and QSR, found a cup in all trials it can be seen that the latter method was able to succeed in half of the time requiring only half of the number of poses.

TABLE II
PERFORMANCE OF THREE DIFFERENT SEARCH METHODS

Search method	Found objects	Average time (sec)	Average poses
random	6/10	68.5	4.8
supporting planes	10/10	33.6	2.3
QSR	10/10	15.6	1.1

However, as we know from previous work, it is sometimes the case that the belief state of the robot does not reflect the actual circumstances of reality. Therefore, we conducted additional variants of the experiment to evaluate how the QSR-based search method performs when the QSR model differs from reality. In variant A the setup and the result is actually the same as from above. In variant B the robot's knowledge is only partially correct as we removed two of the three cups from the environment. That is, in the three locations at which the the QSRs predict cups, a cup is only present at one. In variant C the robot's knowledge is completely misleading as we moved the last remaining cup to a location that is not indicated by any of the three QSRs. The results are shown in Table III.

TABLE III
PERFORMANCE OF QSR-BASED SEARCH UNDER FALSE BELIEFS

Variant	Found objects	Average time (sec)	Average poses
A (correct QSRs)	10/10	15.6	1.1
B (partially correct QSRs)	8/10	55.0	3.1
C (misleading QSRs)	6/10	65.0	3.2

The fact that the robot is still able to find cups in the environment although the QSR model is misleading (variant C) can be explained by the sampling strategy used within our approach. Although views with respect to the QSR model are in general preferable, sometimes none of the sampled navigation goals are directed towards a QSR-related location. In this case, a location with a view cone in direction to a supporting plane gets selected and explored instead.

IV. RELATED WORK

Active visual search has become a popular topic in mobile and service robotics recently. Work done by Aydemir, Sjöö and others in the CogX project ([3], [8], [4]) introduced the sampling-based approach using object location probability distributions. This approach provides and effective and flexible approach to active visual search which is not restricted by the complexity of optimal object search in the general case [9]. The CogX work [4] used the spatial relations "in" and "on" to define object targets. We go beyond this work by using more restrictive spatial models to provide more tightly defined viewing probabilities. As shown in III, increasing search performance in the case where the environment is a close match for these models. Other recent work on object search has tackled larger scale space but used predefined view cones within rooms [10], or has allowed searching over rooms or scenes for unknown objects without constraining their location in 3D [11], [12] such as we are.

In future work we intend to learn positional spatial models from the robot's experience of an environment, creating both environment general spatial models (i.e. keyboards can be found in front of monitors) and more specific models (i.e. the keyboard in Room 133 is often to the right of the monitor). To create such models we can draw on existing work which quantifies qualitative knowledge making it appropriate for our approach (e.g.spatial models [13], [14] or conceptual knowledge [15], [16]), or learns metric object location predictors from experience [12], [17].

V. DISCUSSION AND CONCLUSIONS

In this paper we described an approach on object search on the basis of QSRs between distinguished landmark objects and a sought object. We explained how symbolic representations of QSRs can be transformed into sub-symbolic information (GMMs) to guide a robot in its search towards the most likely object locations. Preliminary experimental results in a robot simulator suggest that QSR-based search methods improve the performance of search tasks when compared to non-QSR-based methods.

However, the transfer of the simulated results onto a real robot platform needs some consideration. First, the semantic camera has to be replaced by an object recognition framework using, for example, the robot's RGB-D sensor. As the recognition rate of the robot's real perception will be lower than that of the semantic camera used in simulation, it might be useful to revisit locations where the robot did not find an object in the first place although the GMMs indicate a high probability. Thereby, the robot could compensate for the false negatives of the perception. This behaviour could also be tested and evaluated in simulation when we induce noise to the semantic camera. A second important aspect concerns the knowledge about landmarks. Ideally, landmarks are be identified and located by the robot itself while operating in an environment. However, it is not clear what kind of objects are good candidates for landmarks as they have to be both salient for the perceptual system and informative with respect to the QSRs with other objects. Therefore we will start with some pre-defined landmarks when we will conduct experiments on the real robot platform.

In future work we would like to learn the currently predefined QSR model over space and time from the real robot's experience. As the robot explores the environment it should collect data about the spatial relations between different types of objects at different times of the day and learn a compact QSR model from it.

With respect to the search method, we plan to sample also 3D view cones at the navigatable goal poses and evaluate them using GMMs and the 3D occupancy map. Using the robot's pan-tilt unit, this would result in even more accurate views at the goal locations and would, for example, make a difference where objects could be located on table-tops or shelves.

Furthermore, it would also be interesting to investigate how the size of the search space, the number of sampled poses, and the weight of QSR models influence each other and to learn to trade-off between the robot's exploration and exploitation behavior.

Overall we belief that the use of QSRs in object search tasks leads to better results in complex and highly dynamic human environments.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 600623, STRANDS, and the EPSRC grant EP/K014293/1.

REFERENCES

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

[2] A. G. Cohn and S. M. Hazarika, "Qualitative spatial representation and reasoning: an overview," *Fundam. Inf.*, vol. 46, no. 1-2, pp. 1–29, 2001. [Online]. Available: http://portal.acm.org/citation.cfm?id=569073.569074

[3] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2818–2824.

[4] K. Sjöö, A. Aydemir, and P. Jensfelt, "Topological spatial relations for active visual search," *Robotics and Autonomous Systems*, 2012, to appear. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889012000851

[5] R. Moratz, B. Nebel, and C. Freksa, "Qualitative spatial reasoning about relative position," *Spatial cognition III*, pp. 1034–1034, 2003. [Online]. Available: http://www.springerlink.com/index/c0a5w5wl7jg177l2.pdf

[6] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013, software available at http://octomap.github.com. [Online]. Available: http://octomap.github.com

[7] G. Echeverria, N. Lassabe, A. Degroote, and S. Lemaignan, "Modular Open Robots Simulation Engine: MORSE," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, 2011.

[8] A. Aydemir, A. Pronobis, M. Göbelbecker, and P. Jensfelt, "Active visual object search in unknown environments using uncertain semantics," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 986–1002, Aug. 2013. [Online]. Available: http://www.pronobis.pro/publications/aydemir2013tro

[9] J. K. Tsotsos, "On the relative complexity of active vs. passive visual search," *Int. J. Comput. Vision*, vol. 7, no. 2, pp. 127–141, Jan. 1992. [Online]. Available: http://dx.doi.org/10.1007/BF00128132

[10] L. Kunze, M. Beetz, M. Saito, H. Azuma, K. Okada, and M. Inaba, "Searching objects in large-scale indoor environments: A decision-thereotic approach," in *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, May 14–18 2012.

[11] D. Joho, M. Senk, and W. Burgard, "Learning search heuristics for finding objects in structured environments," *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 319–328, 2011.

[12] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 2168–2173.

[13] M. Brenner, N. Hawes, J. Kelleher, and J. Wyatt, "Mediating between qualitative and quantitative representations for task-orientated human-robot interaction," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*, Hyderabad, India, 2007, pp. 2072–2077.

[14] C. Burbridge and R. Dearden, "Learning the geometric meaning of symbolic abstractions for manipulation planning," in *Proceedings of Towards Autonomous Robotic Systems (TAROS)*, no. 7429. Springer, 2012, pp. 220–231.

[15] M. Hanheide, C. Gretton, R. Dearden, N. Hawes, J. Wyatt, A. Pronobis, A. Aydemir, M. Göbelbecker, and H. Zender, "Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI'11)*, Barcelona, Catalonia, Spain, July 2011, pp. 2442–2449.

[16] M. Samadi, T. Kollar, and M. M. Veloso, "Using the web to interactively learn to find objects," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, J. Hoffmann and B. Selman, Eds. AAAI Press, 2012.

[17] A. Aydemir and P. Jensfelt, "Exploiting and modeling local 3d structure for predicting object locations," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012.