Using Qualitative Spatial Relations for Indirect Object Search

Lars Kunze* and Keerthi Kumar Doreswamy* and Nick Hawes*

***Abstract*— Finding objects in human environments requires autonomous mobile robots to reason about potential object locations and to plan to perceive them accordingly. By using information about the 3D structure of the environment, knowledge about landmark objects and their spatial relationship to the sought object, 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 *in-front-of*. On the basis of QSRs between landmarks and the sought object we generate metric poses of potential object locations using an extended version of the ternary point calculus and employ this information for view planning. Preliminary results show that search methods based on QSRs are faster and more reliable than methods not considering them.

I. INTRODUCTION

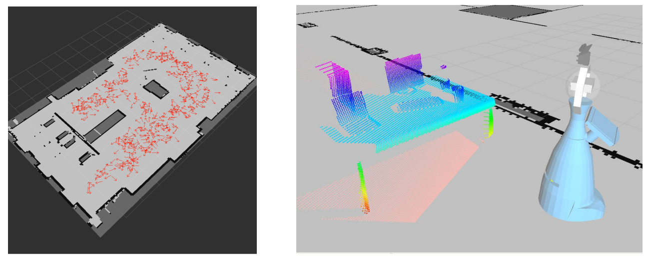
In recent years, we have seen substantial progress towards personal robot assistants performing everyday tasks in human environments [1], [2]. The ability to *search for objects* is an integral part of many of those tasks, as the locations of task-relevant objects are often unknown beforehand, or change over time. Therefore, robots cannot generally assume that an object will be in the same location where it was previously perceived. Hence robots will often have to search for objects in their task environment. The problem of optimal *active-visual search* for objects is NP-hard [3], hence robots need to rely on additional task-relevant information in order to find objects efficiently.

Due to the natural dynamics of human environments, the locations of some objects often change, e.g. pens, mugs, books etc. will be picked up, used, and possibly put down somewhere different. While this occurs, the locations of other objects are generally constant, e.g. monitors, TVs, desks, cabinets etc. do not tend to move in everyday usage. In this paper we present an approach which allows a mobile robot to exploit the location stability of the latter type of objects in order to more efficiently find objects of the former. We are motivated by the knowledge that as research on *long-term autonomy* allows longer and longer robot runtimes [4], robots will be able to exploit the additional experience available to it to learn which objects in its environment do, or do not, tend to change position over time.

In this work, we call objects that do not change their position qualitatively over time *landmarks*, and those that change their position on a regular basis simply *objects*.

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 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 the view points it should use in order to perceive the object. The refinement of the areas considered for active visual 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 [5].

In this paper we investigate how indirect search using *Qualitative Spatial Relations* (QSRs, [6]) 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 objects based on QSRs. Fig. 1 illustrates the questions a robot has to answer when searching for an object, namely, where to stand and where to look. We answer these questions on the basis of QSRs. Starting from the approach of Aydemir et al. [7], we assume that the robot has the following information at the beginning of the search: a 2D map, and a 3D map. To this we add the assumption that the robot also knows the poses of a set of known landmark objects, and a set of QSRs between them and other objects. The *novel contribution* of this paper showing how these additions can be use during indirect search, and how these additions improve object search performance in both simulated and real robot trials.



(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 for objects in human environments requires an autonomous mobile robot to reason about where to stand and where to look.

*Intelligent Robotics Lab, University of Birmingham, United Kingdom {l.kunze|kkd236|n.a.hawes}@bham.ac.uk

The remainder of the paper is structured as follows. In Section II, we put this work into context to related approaches. Section III presents the underlying representations of QSRs, explains how we generate potential object locations based on a ternary point calculus, and shows how we use the resultant information in object search. In Section IV, we present experimental results of different search methods employed in simulated and real object search experiments before we conclude in Section V.

II. 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 ([7], [8], [9]) 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 [3]. The CogX work [9] 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 IV, 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 [2], [15]), or learns metric object location predictors from experience [12], [16].

III. QSR-BASED INDIRECT OBJECT SEARCH

A. Qualitative Spatial Relations (QSRs)

The general approach of QSR-based indirect object search has already been introduced in the previous section: 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 the different types of objects. We refer to objects as *static* if they are only subject to rare and minor location changes over time (i.e. nothing that would qualitatively change their pose); and we refer to objects as *dynamic* if they are subject to frequent location changes (i.e. their pose change qualitatively). In our approach we use *static* objects as landmarks for finding the locations of *dynamic* objects on the basis of QSRs.

In [9], the authors use *topological* relations such as "in" and "on" to specify potential object locations. In this work

TABLE I
LOCATION CHANGES OF OBJECT TYPES

Object type	Location changes	Examples
static	rare and only quantitative	PC, monitor, printer
dynamic	frequent and qualitative	keyboard, mouse, cup

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

we go beyond this by using directional (*left-of, right-of, in-front-of, behind-of*) and distance relations (*close-to, distant-to*) to describe the QSRs between objects. Fig. 2 shows different geometrical configurations of a desktop scene with the same QSRs. In this example, the monitor functions as the landmark and the scene is defined by the following QSRs:

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

To utilize QSR scene descriptions within object search tasks, the qualitative, relational description is transformed into a sub-symbolic, quantitative representation that can directly be integrated with the robot's environment model.

To this end, we use the qualitative positional calculus based on ternary relations [17] that has been developed in the context of robot navigation. The three positions in the calculus are referred by *origin*, *relatum* and *referent*. In this work, *origin* corresponds to the position of the robot, *relatum* to the landmark, and the *referent* to the sought object. In the following we denote these positions by *robot*, *landmark*, and *object*. *Robot* and *landmark* define the reference axis which partitions the surrounding space. Then, the spatial relation is defined by the partition in which *object* lies with respect to the reference axis. To determine the partition, i.e. the spatial relation, we calculate the relative angle ϕ_{rel} as follows:

$$\phi_{rel} = \tan^{-1}\frac{y_{obj} - y_{land}}{x_{robj} - x_{land}} - \tan^{-1}\frac{y_{land} - y_{robot}}{x_{land} - x_{robot}} \quad (1)$$

ϕ_{rel}, is the angle between the reference axis, defined by *robot* and *landmark*, and the *object* point (Fig. 3).

In our work, we assume that the robot position (*robot*), is located in front of the office desk, facing the intrinsic front side of the monitor (*landmark*). That is, contrary to the ternary point calculus we assume that landmark objects are always represented with their full pose. Given this idealized situation, the robot hallucinates potential object locations of, for example, a cup on the basis of QSRs. That is, the robot

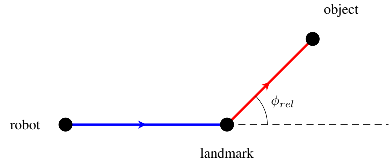


Fig. 3. The relative angle ϕ_{rel} is defined by the reference axis, which is specified by *robot* and *landmark*, and the *object*. The example above illustrates a situation where the *object* is left and behind of the *landmark*.

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 positions for the proximal relations between objects we used a similar approach. The relative radius is calculated as the ratio of the distance between *object* and *landmark* and the distance between *landmark* and *robot*. If the ratio is smaller than a threshold the relation is classified as *close*, otherwise as *distant*. For sampling a distance between objects we use again Gaussian distributions for representing *close* and *distant*. These distributions already take into account that we deal with table-top objects.

Having sampled a number of potential object positions we represent them by a single multivariate Gaussian distribution relative to the landmark. With this approach we could also represent disjoint QSR descriptions. For example, the relations *left-of(Cup,Keyboard)* and *right-of(Cup,Keyboard)* can be represented by two multivariate Gaussians whereby the distributions are weighted by the number of observations a robot made. The overall set of QSRs is then represented by a mixture of 2D Gaussians.

B. Search Method

To perform search, we assume the robot has the following information: a 2D map, a 3D occupancy map, a set of landmark objects, and a mixture of Gaussians generated from a QSR scene description as explained above.

When the search is started, the robot first receives the latest version of the 3D map, calculates the average normal for each voxel, 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 $v_1 \ldots v_m$. Fig. 4 shows the complete 3D Octomap [18] of the environment (left) and the extracted supporting planes according to the averaged normals (right). This is similar to considering all objects to be "on" a supporting plane [9].

In a second step, the robot samples n reachable poses from the 2D map which are denoted by Ψ. At each of these poses, we calculate a 2D view cone according to the robot's sensor specification. The view cones are evaluated with respect to projected 3D occupancy map. To evaluate the

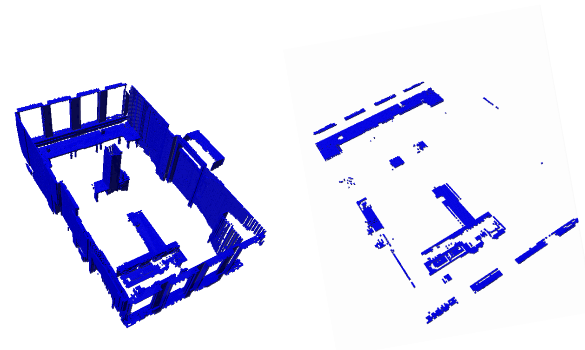


Fig. 4. Left: 3D Octomap of the environment. Right: Extracted supporting planes on the basis of a normal estimation.

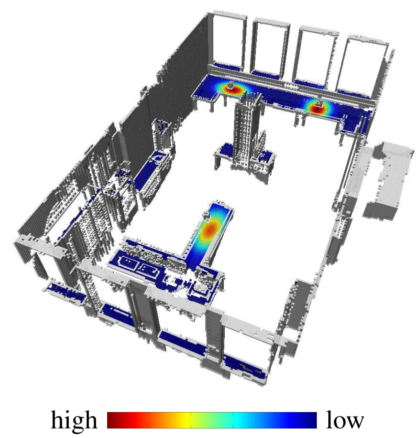


Fig. 5. Voxels of the supporting planes are weighted by the Gaussian mixture model derived from the QSRs

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. 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:

$$\underset{\psi \in \Psi}{\operatorname{argmax}} \sum P_{QSR}(v_i|\omega, \Lambda) In(v_i, Viewcone(\psi)) \qquad (2)$$

where $P_{QSR}(v_i|\omega, \Lambda)$ denotes the probability distribution for voxel v_i given an object type ω and a set of landmarks Λ. The set of landmarks Λ include both their object types and their poses. Fig. 5 visualizes an example of such a probability distribution over the voxels classified as supporting planes on the basis of the Gaussian Mixture model (GMM) 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.

Fig. 6 and Fig. 7 visualize the search progress and 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

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

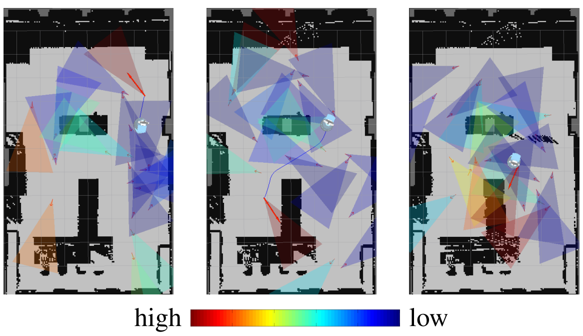


Fig. 6. Viewcone evaluation on the basis of a uniform distribution with respect to supporting planes. Search over three poses.

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 Fig.5.

The search method described above is used to determine the next best view of the robot. In a second step we determine the best view in 3D by evaluating different 3D view cones (or frustums). Fig. 8 visualizes nine frustums from the pose of the robot. Similar to the 2D view cone evaluation, the voxels of the supporting planes are counted and weighted. We use the same equation as above whereby the In function is replaced by a function $In_{3D}(v_i, Frustum(\psi))$. We use the information to actively control the pan-tilt unit of the robot to increase the probability to find the sought object.

IV. EXPERIMENTAL RESULTS

The QSR-based search method has been implemented and evaluated in a simulated and a real environment.

A. Simulated experiments

We used the open source robot simulator MORSE [19] for simulating the environment[1], the SCITOS G5 robot platform[2]

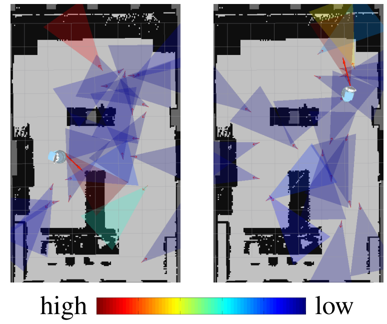


Fig. 7. View cone evaluation on the basis of QSR-based models.

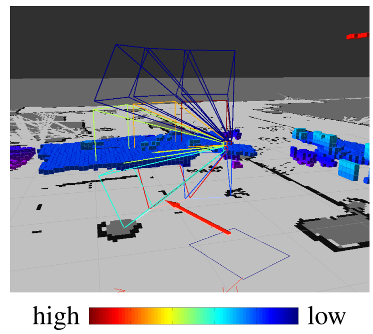


Fig. 8. Viewcone evaluation in 3D.

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:

- In the *random* method 20 locations are sampled from the 2D map. Then, a single goal location is randomly selected and sent to the robot's navigation routine.
- In 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.

[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/`

- In the QSRs-*based* method 20 locations are sampled from the 2D map and evaluated with respect to the 3D voxels weighted according to the QSR-based GMM.

Table II summarizes the results of ten searches using each search method in which only the 2D view cones were used. In total we placed three cups in the environment in locations which are valid according the QSR description given to the robot. 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 were aborted. Please note, that the average time and the average number of searched poses was only calculated on the basis of successful trials. The second noteworthy aspect of the results is how the average time and average number of searched poses decreased when 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 locations which are indicated by the QSR models. In variant C we moved the remaining cup to a different, non-QSR location. That is, the robot's knowledge is wrong as the QSR model is not pointing the location of the cup, but to different locations in the environment. 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 (wrong 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 wrong (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.

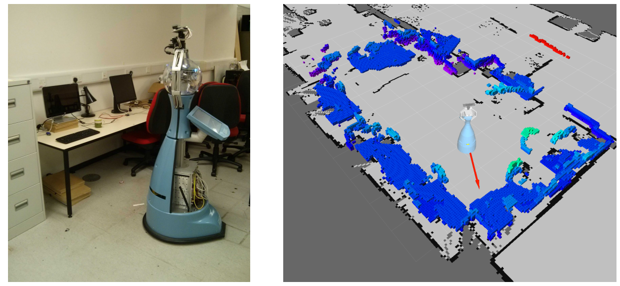


Fig. 9. Left: Robot perceives an office desk. Right: Extracted supporting planes mapped using Octomap.

Table IV shows results whereby the 2D and 3D evaluation of views had been combined. The 2D view cone evaluation used slightly bigger view cone (60 degrees) as the pan-tilt unit can cover a greater area. In the 3D evaluation nine frustums had been assessed at the best 2D pose. The average number of poses almost resembles the result from the 2D experiment. However, the average times increase as at each pose different pan-tilt configurations had been applied.

TABLE IV

PERFORMANCE EVALUATION OF SEARCH USING 3D VIEW CONES

Search method	Found objects	Average time (sec)	Average poses
supporting planes	9/10	69.5	2.2
QSR	10/10	33.4	1.1

B. Real world experiments

We also performed experiments on a SCITOS G5 platform (Fig. 9, left). The robot is equipped with an Asus Xtion Pro Live camera which is mounted on a pan-tilt unit. Similar to the simulated experiments, the task of the robot was to find a cup in an area of our robot lab ($40m^2$). For recognizing objects in the environment we used the an object recognition framework based on 3D CAD models [20] which is integrated in PCL[5]. For recognizing cups (mugs) we trained a classifier based on 50 object categories. Fig. 10 shows some results of the classifier. During the experiments we observed two false positives and a couple of false negatives.

In each view planning step we evaluated 20 samples of 2D poses and nine frustums at the best 2D pose. At each pose the robot actively perceived the scene using the four best frustums. The results of the experiments are shown in Table V. In the *supporting planes* condition the cup was not found in two trials and the search was aborted in other two trials because of false positives. Overall the QSR-based search performed better than the supporting planes method.

[5]Point Cloud Library: `http://pointclouds.org`

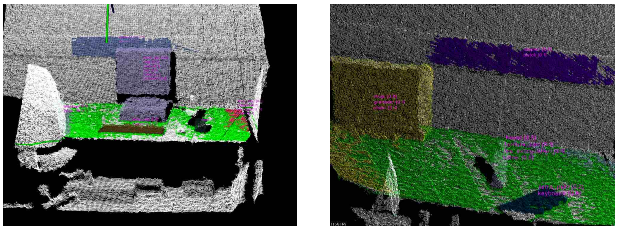


Fig. 10. Object recognition on different desks. Left: True positive of a cup (to the right of the keyboard). Right: False Positive. The monitor was classified as a cup.

TABLE V
PERFORMANCE EVALUATION OF SEARCH USING 3D VIEW CONES

Search method	Found objects	Average time (sec)	Average poses
supporting planes	6/10	149.2	2.6
QSR	9/10	125.9	2.5

V. CONCLUSIONS

In this paper we described an approach on object search on the basis of QSRs between landmarks and a sought object. We explained how symbolic representations of the QSRs can be transformed into sub-symbolic information a robot can use to guide its search towards the most likely object locations. Experimental results in simulation and reality suggest that QSR-based search methods improve the performance of search tasks when compared to non-QSR-based methods.

In future work we would like to learn QSR models over space and time from the 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 day and learn a compact QSR model from it. In this context we will also explore the role of the robot's position with respect to directional spatial relations between objects.

Further, we plan 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 the results show that QSR-based search methods can improve the performance in human environments.

ACKNOWLEDGMENT

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] M. Veloso, J. Biswas, B. Coltin, S. Rosenthal, T. Kollar, C. Mericli, M. Samadi, S. Brandao, and R. Ventura, "Cobots: Collaborative robots servicing multi-floor buildings," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5446–5447.

[2] 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.

[3] 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

[4] E. Marder-Eppstein, E. Berger, T. Foote, B. P. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *International Conference on Robotics and Automation*, 2010. [Online]. Available: http://www.ros.org/wiki/Papers/ICRA2010_Marder-Eppstein

[5] 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.

[6] 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

[7] 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.

[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] 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

[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. 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.

[16] 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.

[17] 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

[18] 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

[19] 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.

[20] W. Wohlkinger, A. Aldoma, R. B. Rusu, and M. Vincze, "3dnet: Large-scale object class recognition from cad models," in *ICRA*. IEEE, 2012, pp. 5384–5391.