Combining Top-down Spatial Reasoning and Bottom-up Object Class Recognition for Scene Understanding

Lars Kunze*, Chris Burbridge*, Marina Alberti†, Akshaya Tippur†,
John Folkesson†, Patric Jensfelt†, Nick Hawes*

***Abstract*— Many robot perception systems are built to only consider intrinsic object features to recognize the class of an object. By integrating both top-down spatial relational reasoning and bottom-up object class recognition the overall performance of a perception system can be improved. In this paper we present a unified framework that combines a 3D object class recognition system with learned, spatial models of object relations. In robot experiments we show that our combined approach improves the classification results on real world office desks compared to pure bottom-up perception. Hence, by using spatial knowledge during object class recognition perception becomes more efficient and robust and robots can understand scenes more effectively.**

I. INTRODUCTION

Accomplishing tasks in human environments can require personal robot assistants to recognise not only individual objects but also multiple objects in a scene. The reasons why understanding a whole scene is occasionally necessary include disambiguating task-related objects (e.g. finding the largest container on a shelf) and distinguishing between different contexts (e.g. determining whether an activity, such as eating or washing up, has started or finished). Given the recent developments of both low-cost depth cameras and software libraries for processing depth images, robot perception systems have improved tremendously over the past decade. Although traditional, bottom-up approaches to robot perception – i.e. those based entirely on information that can be extracted from their sensors – allow a robot to recognise objects, they also have their limitations, for example, in situations where objects are partially occluded. Under these and similar circumstances, background knowledge about the typical spatial relations between objects (e.g. that a keyboard usually appears in front of a monitor) can help a robot to recognise or categorise an object reliably, even when perception is uncertain. Within the STRANDS project[1] we are developing service robots which can patrol and observe indoor environments for weeks and months at a time. During these patrols, the STRANDS robots will regularly observe the same types of objects in a variety of arrangements in their environment, e.g. desks featuring mugs, laptops, books etc. We are developing approaches to improve the overall performance of robot perception which take this experience into account.

*Intelligent Robotics Lab, School of Computer Science, University of Birmingham, United Kingdom bob@cs.bham.ac.uk

†Centre for Autonomous Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden rosie@kth.se

[1]http://strands-project.eu

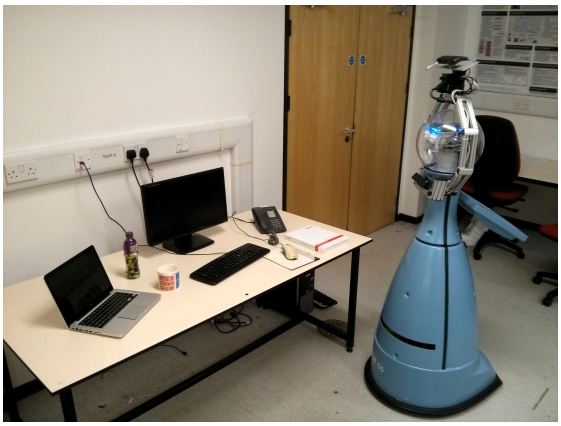

Fig. 1. Robot perceives a scene of an office desk.

Let us consider the scene in Figure 1. The robot's object categorisation component may report a *Mug* on the right of the *Keyboard*. However, given the arrangement of the objects, spatial reasoning could tell the robot that the object classified as *Mug* is more likely to be a *Mouse* given its location to the right of the *Keyboard*. In this work we have created a system which does this by integrating a 3D perception system for object classification with a spatial reasoning component. We have created a framework which combines the output of the perception system with learned spatial models that predict the class of an object given its relations to other objects. In this framework we have explored two possible representations for learning spatial features, one based on metric spatial properties, and one based on qualitative spatial relationships. We have evaluated both representations on real robot data. The main contribution of this work are:

- a framework for unifying perception and reasoning components,
- a comparison of spatial representations in this framework,
- a collected data set of classified office desk scenes, and
- an experimental evaluation and analysis of these scenes.

The remainder of the paper is structured as follows. First, we discuss related work in Section II. Second, we describe the two main components of our approach: bottom-up perception (Section III) and top-down reasoning (Section IV). Third, we explain the setup of the experiments and present their results in Section V.

II. RELATED WORK

Object co-occurrence is a simple way to provide context to perception tasks. Examples include simple object co-occurrence statistics in class-based image segmentation [1]; the use of object presence to provide context in activity recognition [2]; and the linking of object presence to room category in semantic mapping [3]. The work is this paper goes beyond these examples by use a richer, more structured representation to encode spatial information in 3D.

Spatial relations have been used previously to provide contextual information to image processing work. For example, a hierarchy of spatial relations alongside image features has been used to support multiple object detections in a single image [4], and spatial relations and contextual information are commonly used in activity recognition from object tracks in video e.g. [5]. These approaches are restricted to 2D image input, whilst we work on 3D scenes (albeit static ones).

Roboticists have used 3D spatial information for semantic pruning in object search problems in human environments [6], [7]; for activity recognition [8]; and conditional object recognition [9], [10]. Our approaches go beyond this work with additional qualitative spatial relations, but our models for encoding 3D spatial context could be applied in these use cases. In addition, we contribute an explicit evaluation of different representations of spatial context (metric vs qualitative) in a long-term autonomy setting.

III. BOTTOM-UP ROBOT PERCEPTION

In this work we build on a model-based object class recognition framework developed by Aldoma et al. [11]. For training the object classifier in this framework we use 3D CAD models from 3DNET [12]. Our extended object categorisation framework performs the following processing steps:

(1) receive a point cloud from the RGB-D sensor
(2) find the largest supporting plane in the point cloud
(3) segment out *object clusters* (potential objects) on the supporting plane
(4) for each object cluster:
  (a) compute its similarity to all object models
  (b) assign a class probability for each recognised object class based on the similarity measure
  (c) add a small probability to all object classes to assure that they all have a non-zero probability and then re-normalise
(5) return a recognition result for all segmented clusters

Given a set of object categories $T_1 \dots T_m$, the recognition result includes the following for n clusters: bounding boxes, $b_1, \dots, b_n$; centroids, $c_1, \dots, c_n$; the most likely labels, $L_1 \dots L_n$; and object class probabilities for each cluster i for each of the m potential object classes: $P_i^1, \dots, P_i^m$.

Figure 2 visualises the segmentation of an example point cloud into several clusters. For each cluster, Table I shows the recognition result as computed in step (4) of our algorithm. The highlighted cells correspond to hypotheses output by the perception system. We see, for example, that cluster O_3, corresponds to a monitor in reality and that the perception system has classified it either as a monitor or a book (step (4b)). To account for imperfections in the uncertainty model in the perception system and to avoid assigning probability 1 (complete certainty) or 0 (impossible) to any category we assign a small fixed probability to the other categories and then re-normalise each column in the table (step (4c)).

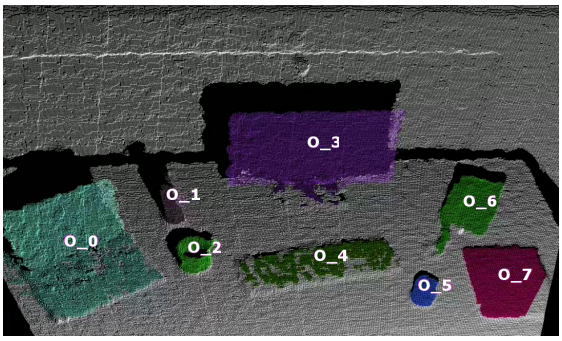

Fig. 2. Segmented clusters on an office desk.

TABLE I
OBJECT CLASS RESULTS FOR FIGURE 2.

Recognized Obj. Class	**Cluster ID (Ground truth)**							
	(Laptop) O_0	(Bottle) O_1	(Mug) O_2	(Monitor) O_3	(Keyboard) O_4	(Mug) O_5	(Telephone) O_6	(Book) O_7
Book	0.03	0.03	0.03	0.25	0.03	0.03	0.03	0.10
Bottle	0.03	0.74	0.03	0.03	0.03	0.03	0.03	0.03
Keyboard	0.03	0.03	0.03	0.03	0.74	0.03	0.03	0.03
Laptop	0.32	0.03	0.03	0.03	0.03	0.17	0.03	0.53
Monitor	0.03	0.03	0.03	0.53	0.03	0.10	0.10	0.10
Mouse	0.17	0.03	0.03	0.03	0.03	0.03	0.03	0.10
Mug	0.32	0.03	0.75	0.03	0.03	0.53	0.03	0.03
Telephone	0.03	0.03	0.03	0.03	0.03	0.03	0.67	0.03

IV. TOP-DOWN REASONING

The following sections present two approaches for learning models of spatial context from observations of object configurations, and then applying these models to influence object classification results from perception. The first approach (Section IV-A) is based on metric information, the second (Section IV-B) abstracts from metric information to purely qualitative relations. We have included both approaches as we expect these representations to have different properties and potentially play different roles in a robot system (e.g. the qualitative models can also be used for grounding language).

A. Metric Spatial Relations

The following approach uses a voting strategy to capture the metric spatial and semantic coherence of object arrangements.

a) Spatial Features and Spatial Relation Based Features: To capture object geometry and the spatial distribution of objects in the scene we use the features proposed in [13]. *Single object features (SOF)* f_{O_i}, where O_i is the i^{th} object, are computed from the 3D spatial characteristics of the object w.r.t. a reference frame (here the table, with the front-left table corner used as the origin of an extrinsic reference system aligned with the two horizontal table axes). The set of features consists of: the length of the object projection along the X, Y and Z table axes; 3D coordinates of the object centroid in the reference system defined by the table; and the horizontal bearing of object centroid from front-left table corner. *Object pair features (OPF)* represent the pairwise spatial distribution of the objects, f_{O_i,O_j} as: Euclidean distance between object centroids; Euclidean distance between centroids in the X-Y plane; bearing between the two object centroids computed in the reference system defined by the table; ratio of object volumes and vertical displacement between object centroids.

b) Learning Spatial Models: In the training phase, a set of models for each of the object class categories are learned by using a Gaussian Mixture Model-based representation to encode the multivariate probability distribution of *SOF*. The relationship of the different object category pairs are modelled by applying a GMM on the multi-dimensional feature space of *OPF* set.

c) The Voting Scheme: In the inference phase, a voting scheme is applied and a score $\mathrm{score_A}(O_i, T_p)$, is computed for the assignment of each test object, O_i, to each of the possible categories, T_p, based on the spatial relations with the reference system and with the other objects and on typical object occurrence and co-occurrence presence. $\mathrm{score_A}(O_i, T_p)$ is computed as the sum of *pairwise scores* that involve the considered assignment:

$$\mathrm{score_A}(O_i, T_p) = \sum_{\substack{j\in\{1,\ldots,n\}\\ j\neq i}} \sum_{\substack{q\in\{1,\ldots,m\}\\ q\neq p}} \mathrm{score_P}\left((O_i, T_p), (O_j, T_q)\right), \quad (1)$$

where n is the number of test objects and m is the number of object categories. The *pairwise score* is defined as:

$$\mathrm{score_P}\left((O_i, T_p), (O_j, T_q)\right) = \mathrm{score_{SO}}(O_i, T_p) \cdot \mathrm{score_{SO}}(O_j, T_q) \cdot \mathrm{score_{OP}}\left((O_i, O_j), (T_p, T_q)\right). \quad (2)$$

$\mathrm{score_{SO}}(O_i, T_p)$ and $\mathrm{score_{OP}}\left((O_i, O_j), (T_p, T_q)\right)$ take into account the likelihood values of the category models and the likelihood value of the category pair model given the extracted features, corresponding to the conditional probability of the features given the trained models. Additionally, the scores integrate, as a-priori weights, the occurrence probability of the individual object categories and the co-occurrence probability of the object category pairs both estimated using frequency counts on the training database, and the confidence of the perception system:

$$\mathrm{score_{SO}}(O_i, T_p) = p(f_{O_i} \mid T_p) \cdot \frac{\max(1, N_{T_p})}{(1 + N_{tot})} \cdot P_i^p \quad (3)$$

$$\mathrm{score_{OP}}\left((O_i, O_j), (T_p, T_q)\right) = p(f_{O_i,O_j} \mid T_p, T_q) \cdot \frac{\max(1, N_{T_p,T_q})}{(1 + N_{tot})}, \quad (4)$$

where N_{T_p} is the number of training scenes where an object of type T_p is present, N_{T_p,T_q} is the number of scenes where object of both T_p and T_q types are present, N_{tot} is the total number of training scenes and P_i^p is the confidence or probability value provided by the perception system that object i is of type T_p. The numerator and denominator terms, $\max(1, N_{T_p})$, $\max(1, N_{T_p,T_q})$ and $(1 + N_{tot})$, ensure that occurrence and co-occurrence weights are never 0 or 1.

B. Qualitative Spatial Relations (QSR)

Qualitative relational approaches abstract away the metric information of a scene and instead represent it using relations predicates such as *left-of* and *close-to*. The approach described below generates these predicates from geometric descriptions, then builds a probabilistic model to reason about the classes of related objects.

1) Qualitative Relations: In this work we adopt a semi-supervised approach to generating spatial relation predicates which combines geometric calculi with clustering methods. This produces a qualitative description constructed from 12 predicates: 4 directional, 3 distance, 3 size and 2 projective.

Directional predicates are created using the *ternary point calculus* [14]. The three positions in the calculus are the *origin*, *relatum* and *referent*. In our work, the *origin* is the position of the robot, *relatum* is a landmark object, and the *referent* is the object under consideration. In the following we denote these positions by *robot*, *landmark*, and *object*. *Robot* and *landmark* define the reference axis which partitions the surrounding space. The spatial relation is then defined by the partition in which *object* lies. In order to determine the partition we calculate the relative angle ϕ_{rel} as follows:

$$\phi_{rel} = \tan^{-1}\frac{y_{obj} - y_{land}}{x_{obj} - x_{land}} - \tan^{-1}\frac{y_{land} - y_{robot}}{x_{land} - x_{robot}} \quad (5)$$

ϕ_{rel}, is the angle between the reference axis, defined by *robot* and *landmark*, and the *object* point. Dependent on this angle we assign directional relations (*behind, in-front-of, left-of, right-of*) to pairs of objects.

Distance relations are determined by clustering the metric distance relations observed into a training set into a given number of clusters, each of which will correspond to a qualitative relation. Based on the membership of a geometric relation to a cluster, the associated qualitative relation is then assigned to the objects it involves. In our technique we use three different relations: *very-close-to, close-to, distant-to*.

Size predicates compare the bounding boxes of two objects. Each axis is compared individually, creating three predicates *shorter-than*, *narrower-than*, and *thinner-than*.

Projective connectivity between two objects uses Allen's interval calculus [15] with the projection of the objects' axis-aligned bounding boxes onto the x or y axis. The *overlaps* predicate is then extracted for each axis.

2) Probabilistic QSR-based Reasoning: Our objective is to infer the types of all objects given a symbolic scene description

$$S = C_1 \wedge C_2 \wedge ... \wedge C_n \tag{6}$$

where C_n is a relation R between two objects O_a and O_b:

$$C_n = (R \;\; O_a \;\; O_b), \tag{7}$$

for example `(shorter-than object15 object7)`.

From a training set of annotated scenes, we use the occurrence count for each relation to estimate the probability it will hold given the object types of its arguments:

$$p(R_n^{ab}|L_a, L_b) = \frac{N_{R_n,L_a,L_b} + 1}{N_{L_a,L_b} + 1} \tag{8}$$

where R_n^{ab} is one of the 12 symbolic relations between two objects O_a and O_b with class labels L_a, L_b, N_{L_a,L_b} is the number of times that objects of types L_a and L_b have co-occurred across all training scenes, and N_{R_n,L_a,L_b} is the number of times that relation R_n has occurred between objects of types L_a and L_b across all training scenes.

Then, given a new scene description S containing object types from perception with a certain confidence, we find all object labels simultaneously. Assuming that relations hold independently, we can apply Bayes theorem recursively to find the labels of all objects:

$$p(L|R_1, R_2...R_n) \propto \prod_{i=1..n} p(R_i|L)p(L) \tag{9}$$

where L is a vector of class labels for the objects in S, and R_i is the ith relation in S. The prior probability of the labels $p(L)$ comes from the robot's perception model:

$$p(L) = \prod_{i=1..n} p(L_n) \tag{10}$$

where all n object class labels are independent and provided with their confidences $p(L_n)$.

Finding the optimum class labelling estimate $\hat{L}$ for the objects is then equivalent to finding the maximum posterior in Equation 9. To avoid computational arithmetic problems when dealing with very small unnormalised probabilities, we replace the product in Equation 9 with a sum of logarithms:

$$\hat{L} = \arg\max_{L} \sum_{i=1..n} \log p(R_i|L) + \log p(L) \tag{11}$$

We perform the maximisation using gradient ascent.

V. EXPERIMENTS

A. Experimental Setup

To test our approaches we recorded a data set of 20 object arrangements on an office desk. The objects used were taken from the following classes: *Book, Bottle, Keyboard, Laptop, Monitor, Mouse, Mug* and *Telephone*. Not all scenes included objects of all classes. The objects were arranged in accordance to our previous observations of real world office desks [16]. Each desktop scene was perceived by a SCITOS G5 robot (Figure 1) from three different views, resulting in 60 scenes. The changes in view changed the labels provided by perception, with each different view varying by roughly 1 or 2 labels from its alternatives. Also, the robot's localization error with respect to the table added noise to the perceived data, making it more variable. After acquiring a point cloud with the robot's RGB-D sensor we ran the object class recognition framework (Section III). Cluster artefacts and under-segmented clusters (e.g. two objects grouped together) were removed manually from the results, then all remaining clusters were labelled with the ground truth.

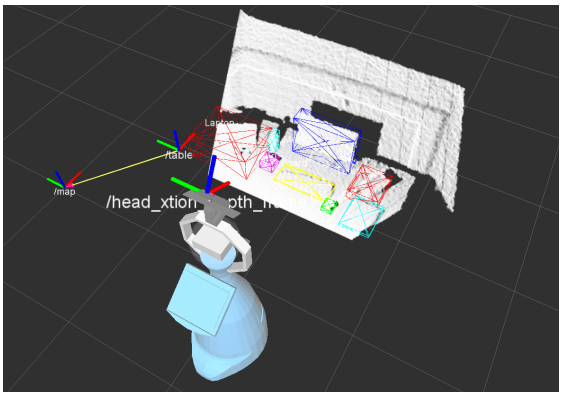

Fig. 3. Localized robot perceives objects on office desk.

Using this setup we performed two experiments with different foldings of the data: leave one out foldings (LOOF) and random foldings (RF). The LOOF experiments evaluate how our approaches operate on unseen desks (the one that is left out) after training on all other desks. This is to replicate the condition of a trained robot encountering a new desk, a likely situation in our long-term scenario. The RF experiment evaluates the overall performance of the our approaches. For the LOOF experiments we split the data desk-wise into training and test sets with 19 and 1 desk(s) respectively (or 57 and 3 scenes). For RF we did 6-fold cross validation leading to training and test sets with 50 and 10 scenes respectively.

For each experiment we ran the robot's bottom-up perception (BUP), plus the spatial metric (SM) and qualitative models described above. For the qualitative approach we explored different combinations of relations, labelled as follows: ternary point calculus (T), qualitative distance (D), relative size (R), projective connectivity (C). Combinations of labels indicate several relations were used (e.g. TDRC uses all relations). We also ran both spatial reasoning approaches using BUP to only segment, but not classify, objects. This was to test whether BUP is really necessary for classification given the prior experience of the robot. Whilst this is not an entirely fair comparison to make (as a visual classifier trained only on the same data as our spatial reasoners would undoubtedly perform as least as well), it does show how much information is captured in the spatial models. To create these perceptionless systems the SM system was modified to fix the perception score weighting for all objects (setting $P_i^p = 1$) and in the QSR system the perception prior ($p(L)$) was replaced with a uniform one.

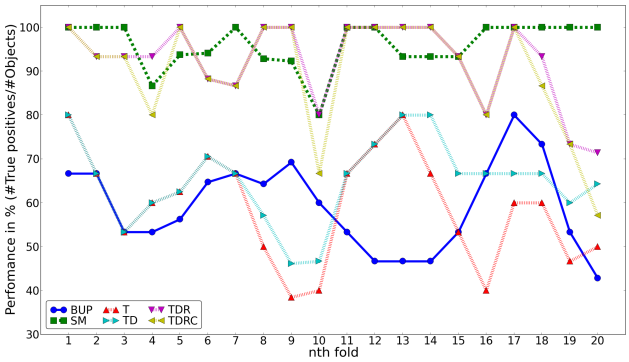

Fig. 4. Individual fold results for the LOOF experiment.

B. Experimental Results

1) Leave-one-out Foldings (LOOF): We measure performance on our task as the percentage of correctly classified objects in a scene. As shown in Table II, all of our combined approaches were, on average, better than bottom-up perception (BUP) on its own. BUP had an average performance of 59.19%. The combined approach that performed best, with an average performance of 95.98%, is the metric approach (SM). This approach used both intrinsic and extrinsic object features. From the qualitative relational approaches, TDR performed best with an average performance of 92.31%. This suggests that considering the projective connectiveness relations (C) do not help disambiguate objects. On the other hand, using relative size relations (R, a is taller than b, c is wider than a etc.) is critical to achieving performance comparable to SM. We investigate this further below.

TABLE II
METHOD PERFORMANCE IN THE LOOF EXPERIMENT

Method	**With Classification**		**Without Classification**	
	Mean	Std. Dev.	Mean	Std. Dev.
BUP	59.20	9.82	-	-
SM	95.98	5.35	95.65	5.30
T	59.24	12.23	45.38	15.24
TD	65.03	9.33	54.72	12.30
TDR	92.32	9.18	90.98	10.32
TDRC	89.94	12.37	88.94	12.14

Figure 4 reports the performances of all approaches on the 20 different object arrangements (when trained on the remaining 19). Whilst performance on individual folds matches the averages well, some object arrangements challenged all systems. For example, fold 10 is a desk where the mouse appeared on the left of the keyboard – an unusual spatial configuration in our data set. This spatial configuration caused the SM and TDR approaches to score only 80%, well below averages of 95.98% and 92.31%. This was due to the relations suggesting the mouse should not be classified as one due to it relatively improbable position.

2) Random Foldings (RF): In the random foldings experiment the methods have been evaluated on a larger test set of arbitrary desks. The average performance of each approach is shown in Table III. Overall, the results are largely similar to those in LOOF. That is, all combined approaches improve the performance of the robot's perception system, with SM making the largest improvement.

TABLE III
METHOD PERFORMANCE IN THE RF EXPERIMENT

Method	**With Classification**		**Without Classification**	
	Mean	Std. Dev.	Mean	Std. Dev.
BUP	60.86	4.28	-	-
SM	93.95	5.39	93.95	5.39
T	65.31	10.16	59.28	9.33
TD	67.54	9.33	60.50	8.20
TDR	88.23	6.33	86.26	6.20
TDRC	87.59	5.87	85.29	5.94

3) Analysis: The results above show that the inclusion of spatial information can significantly improve the performance of a desktop object classification system, compared to just using bottom-up perception. The results from running the spatial reasoning systems without the classification results from perception (right hand side of Tables II & III) show that, at least on the dataset we tested on, spatial information alone may be enough to correctly classify most objects, provided the correct spatial features are used. In these results, removing BUP decreases performance only marginally for all methods but T and TD. These latter methods rely solely on relative angles and distances between objects whereas SM, TDR and TDRC also contain information about (relative) size and shape. While crude in nature this additional information makes up for the information from BUP in this case.

As described above, encoding object size is essential to achieving good performance on our data. This is included as the R (relative size) relation in the QSR approaches, and as both single and pair features in SM. In fact, running the QSR approach with just the R relation achieves 94.6% (σ 5.24) and 89.1% (σ 6.52) on the LOOF and RF experiments respectively, out-performing all other relational approaches. In our dataset we did not vary object instances when we varied object presence and arrangement, thus size became strongly discriminative: a mug is always smaller than a monitor regardless of position. This result could be interpreted in one of two ways. The first way is as a weakness in our methodology: the BUP classification system is designed to cope with variation in both in appearance and size and was trained on 3D CAD models, rather than the instances in our data, thus its standalone performance should not be compared to a system which can (successfully) overfit on training instances. If the spatial models were trained to include a greater degree of variability in object classes, we should expect to see reduced performance on some specific instances (but greater ability to generalise).

The alternate interpretation is that spatial models, when correctly trained, can supplant perception in some tasks. Object instance size does not vary a great deal within certain classes, and thus exploit this feature can provide a performance gain. This can be supported by data. Figure 5

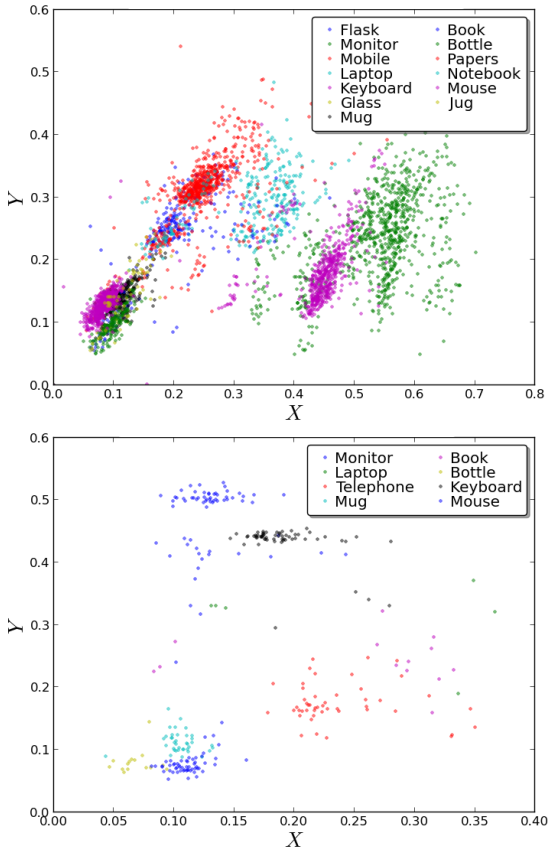

Fig. 5. Object bounding box sizes. Top: 3783 human-segmented objects from a large dataset. Bottom: 303 BUP-segmented objects from our data.

(Top) shows the 2D (table plane) dimensions of 3783 object instances drawn from 13 classes in a large database of 3D desktop scenes we have recently collected. These are naturally occurring scenes and thus contain natural variation within classes. This shows that many object classes have relatively consistent sizes, and that these dimensions can be used to discriminate classes. Therefore relative size is likely to be a useful discriminative feature for real scenes. This will be the same as for any spatial relation which does not vary *qualitatively* across training and test data.

The caveat to this is that any spatial abstraction must be chosen such that it is robust to sensor or interpretation noise. Figure 5 (Bottom) shows the bounding box dimensions of the objects which were automatically segmented by bottom-up perception for the dataset in this paper. Despite every class instance being an observation of an identical object, there is significant variation between resulting bounding box sizes. For objects from different classes which are close in size, this variation could result in the feature become less useful. One approach to dealing with this would be to include a feature selection step prior to training [17].

VI. CONCLUSIONS

In this paper we presented metric and qualitative spatial reasoning techniques that provide top-down guidance to a bottom-up object recognition framework. To compare these approaches, we acquired a data set of desktop scenes using the perception system of a robot. Using this data we demonstrated that such reasoning approaches can improve the overall performance of a perception system. In future work we plan to deploy the developed techniques on our STRANDS robots within a long-term scenario. In this context, challenges include how robots can adapt their spatial, relational models over time, and how they can learn predictive models without requiring prohibitive amounts of supervision. We will also explore how we can use spatial reasoning to replace the manual processing we had to do to clean up the segmentation errors in our dataset. Overall, the developed methods we have presented are useful in perception systems of robots which can exploit their knowledge and experience about spatial structures and contexts in their environments.

REFERENCES

[1] L. Ladicky, C. Russell, P. Kohli, and P. Torr, "Inference methods for crfs with co-occurrence statistics," *International Journal of Computer Vision*, vol. 103, no. 2, pp. 213–225, 2013.

[2] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei, "Objects as attributes for scene classification," in *Trends and Topics in Computer Vision*, K. Kutulakos, Ed., 2012, vol. 6553, pp. 57–69.

[3] M. Hanheide, C. Gretton, R. Dearden, N. Hawes, J. L. Wyatt, A. Pronobis, A. Aydemir, M. Göbelbecker, and H. Zender, "Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour," in *IJCAI*, Barcelona, Spain, 2011.

[4] M. J. Choi, J. Lim, A. Torralba, and A. Willsky, "Exploiting hierarchical context on a large database of object categories," in *CVPR*, 2010, pp. 129–136.

[5] K. S. R. Dubba, A. G. Cohn, and D. C. Hogg, "Event model learning from complex videos using ilp," in *ECAI*, 2010, pp. 93–98.

[6] A. Aydemir, K. Sjoo, J. Folkesson, A. Pronobis, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *ICRA*, 2011, pp. 2818–2824.

[7] L. Kunze, K. Kumar, and N. Hawes, "Indirect object search based on qualitative spatial relations," in *ICRA*, Hong Kong, China, 2014.

[8] M. Karg and A. Kirsch, "Acquisition and Use of Transferable, Spatio-Temporal Plan Representations for Human-Robot Interaction," in *IROS*, 2012.

[9] T. Southey and J. J. Little, "Learning qualitative spatial relations for object classification," in *IROS 2007 Workshop: From Sensors to Human Spatial Concepts*, 2007.

[10] A. Kasper, R. Jakel, and R. Dillmann, "Using spatial relations of objects in real world scenes for scene structuring and scene understanding," in *ICAR*, 2011.

[11] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, and S. Gedikli, "Using the point cloud library for 3d object recognition and 6dof pose estimation," *Robotics & Automation Magazine*, vol. September 2012, p. 12, 2012.

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

[13] M. Alberti, J. Folkesson, and P. Jensfelt, "Relational approaches for joint object classification and scene similarity measurement in indoor environments," in *AAAI 2014 Spring Symposia: Qualitative Representations for Robots*, 2014.

[14] R. Moratz, B. Nebel, and C. Freksa, "Qualitative spatial reasoning about relative position," *Spatial cognition III*, pp. 1034–1034, 2003.

[15] J. F. Allen and L. F. Allen, "Maintaining knowledge about temporal intervals," *Communication of ACM*, pp. 832–843, 1983.

[16] L. Kunze, C. Burbridge, and N. Hawes, "Bootstrapping probabilistic models of qualitative spatial relations for active visual object search," in *AAAI Spring Symposium 2014 on Qualitative Representations for Robots*, Stanford University in Palo Alto, California, US, 2014.

[17] J. Young and N. Hawes, "Effects of training data variation and temporal representation in a qsr-based action prediction system," in *AAAI Spring Symposium 2014 on Qualitative Representations for Robots*, 2014.