Object Persistence in 3D for Home Robots

Parnian Alimi and David Meger and James J. Little

Abstract— **This paper proposes a practical method to achieve object persistence – that is, a memory of the objects that have previously been discovered by a system, along with a way to update this memory as objects move. Our approach creates and maintains an object representation that includes 3D location, pose, semantic label, and appearance from multiple viewpoints. We employ a change-based attention operator which allows for efficient re-recognition of moved objects without wasted computation on stationary items. Our global model incrementally learns object appearances and which allows for increasingly successful persistence tracking as more time is spent in the environment. Our results demonstrate that the system is able to successfully maintain its model as objects in a cluttered kitchen area are moved during a series of days.**

I. INTRODUCTION

Objects are the highlight of a home robot's interactions with the world. They are the items to be grasped, they ground the meaning of terms in interaction with humans, and they are necessary for nearly every meaningful task performed by a home robot. However, most modern robots lack the ability to detect, track, and reason about the large number of objects that are likely to occur in modern homes. This is due in part to the challenges in describing and re-detecting objects once they have been demonstrated to a system. But, perhaps more fundamentally, it is because only a few systems have been proposed for chronicling the information provided by the user and from an object recognition module.

Consider a human's view of the objects in their home. Although we are highly competent at instantaneously recognizing all objects we actually look at, we often know where many things are without even looking, because we have previously performed successful detection, and we are confident that nothing has changed. This paper considers a similar paradigm for a mobile robot attempting to keep track of a large number of objects in a home. We call this approach *semantic object persistence* – where meaningful labels are attached to items in the world and where the location of those objects changes over time.

We describe an *object persistence* system that is interested only in the changes that it detects in the environment. These changes become seeds to query the users about for labels: "What is the name of this object that has moved?" Once labels have been given, the system continues to explain the subsequent changes and updates its persistent object model. This is beneficial for a number of reasons. First, it drastically reduces the computation required to continuously track the position of a large number of objects, which is a primary practical limiting factor that currently restricts the number of objects simultaneously available to a system. Second, the fact that we explicitly expect objects to move and that our system deals with this robustly is promising as an add-on to methods such as SLAM which make the "static world" assumption. Our system is able to generate a list of changed objects that can be ignored by such an approach.

Beyond change detection, a key portion of our method is the verification and update of the *semantic* object labels when changes occur in the environment. Consider an example scenario where the robot is taught about a set of objects by the user and leaves the room until the next day (n.b. throughout the paper we explain *visits* to the same scene as *days* to give the reader the intuition but our approach can handle any frequency of repeated visits to a place). The objects in a scene will often have moved when a robot returns. The task is to correctly update the robot's belief of the object locations and labels at the end of each *day* after detecting changes, performing recognition and explaining changes that might include an object moving to a new location, an object being removed from the scene, or a new object being added to the scene.

Our system depends on the reliable 3D data that is now available to many home robots, such as from an RGB-depth sensor like the Microsoft Kinect. This 3D data allows for highly accurate change detection, which correctly guides our approach only to update the state of objects that have moved. Also, appearance signatures derived from high resolution 3D sensory data are highly repeatable and this allows for successful tracking of the semantic labels of regions as objects are shifted, rotated, added and removed from a scene.

The following section describes related work in the field. Section III continues by describing the technical details of our approach for object persistence. We then present qualitative and empirical evaluation of the system to demonstrate its performance in realistic scenes, and finally present conclusions.

II. RELATED WORK

Many previous authors have proposed robotic systems capable of discovering and reasoning about semantic objects, especially within home environments. Perhaps the most similar work to ours is in unsupervised object discovery. Herbst *et al.* [1] have used 3D data from a Kinect-like sensor to attempt to locate objects that have moved between frames. [2] performs a similar task based on laser-rangefinder data. Unlike our method, these objects are not assigned labels by the user, and the authors do not form a reliable multi-view

All authors are with the Department of Computer Science, University of British Columbia, Vancouver, Canada {parnian,dpmeger,little}@cs.ubc.ca

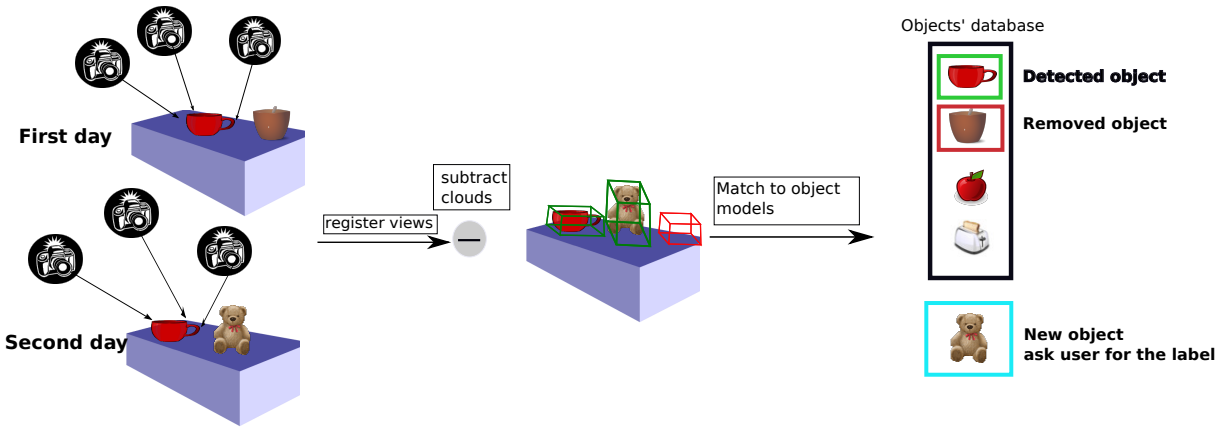


Fig. 1. A diagram of the interactions between components in our object persistence method.

object model as is done in our work. However, we have been inspired by each approach.

Several methods for object search have explored the usefulness of global models or priors on object locations. Kollar *et al.* [3] build a probabilistic model of object locations using captions on Flicker photos. Given the pre-built map of the building and locations for a subset of objects they compute the optimal path to explore the space for query object. Meger *et al.* [4] described a method that used image-based visual saliency to propose promising regions, previously learned object models to label those regions, and a SLAM framework to track detections over time. However, neither of these approaches describe how to update object models once they are acquired, which is a key component to avoid the computation needed to continuously re-recognize all objects. This concept is similar to so-called "lifelong mapping", which was first fully defined in [5]. Finally, an approach for representing and updating semantic information about an environment is given by Hawes *et al.* [6]. This work considers larger scale spatial information and focuses on selection of behaviours rather than focusing on updating a perceptual model, but we are inspired by several of their concepts.

Our work builds appearance models based on features derived from sensed 3D geometry. Many features of this type have been proposed. The most successful among early attempts is spin images [7] which computes a shape descriptor using the relative orientation of the normals. More recently, Viewpoint Feature Histograms (VFH) [8], Fast Point Feature Histograms (FPFH) [9] and Normal Aligned Radial Features [10] have all been proposed and evaluated on high quality data from textured stereo or Kinect-like devices. We note that excellent matching performance has been demonstrated by VFH, in particular when an object is viewed from the same vantage point at both training and test time (e.g. the results of the Recognition Infrastructure (ReIn) [11] by Muja *et al.*) This has motivated our selection of the VFH feature for our work.

III. OBJECT PERSISTENCE METHOD

At a high level, our system is composed of:

1) A geometric registration system that determines how the current sensor position differs from the position at which the scene was originally perceived
2) Geometric change detection that leverages the registration information to find objects that have likely moved during the period between observations
3) A multi-view 3D object modeling approach that can be incrementall updated as new views of an object are located and which can assign semantic labels to each observed item in the world
4) A high-level object persistence reasoning module that combines the outputs of the previous three components to maintain a complete scene representation and add user information using queries when this is needed.

As depicted in Figure 1 system components 1) and 2) act as an attention operator for our system, focusing later computation and allowing the system to quickly ignore large parts of the world which are unchanged, which greatly improves efficiency. The system analyzes the changed regions with module 3) based on 3D histogram descriptors and an efficient nearest-neighbours classifier. When a new object is located, the user is queried for its name. When a change is recognized as being an object that was previously known, the model for that object is updated. All of the previously seen objects that are unchanged can simply persist in the model, allowing for their use in tasks without additional perception effort. Finally, in some cases an object has simply been removed from the scene, so the corresponding object model is removed from the representation.

The remainder of this section will provide additional details about each component of the method.

A. Registration and Change Detection

Our system collects 3D point clouds each time it visits a scene. In order to reduce complexity and focus the attention of the remainder of the processing, we efficiently detect only those parts of the environment that have changed since the last visit. The first step in identifying changes is to register the current observed geometry to that which was observed previously. This step necessarily involves solving for the

transformation between the sensor positions in each case. We note that this registration information is often available to a robot system, such as from its SLAM module, or based on visual structure-from-motion (SfM) information. In our experimental evaluation, we have not depended on any such system. Rather, we perform point-to-point Iterative Closest Points (ICP) [12] directly on the sensed 3D point clouds in order to determine the sensor motion that has occurred.

Upon obtaining registered sensor information, we search for regions of the world that have changed since they were last viewed. To avoid errors due to sensor discretization, we down-sample the 3D data from each view into a voxel-grid data structure. The difference between occupied voxels gives the regions of space which are occupied on one day and not the other. We then cluster these changes into discrete groups using the Euclidean Clustering algorithm described in [13]. This approach allows for further filtering of the data by discarding small clusters, but more importantly, it allows the remainder of the algorithm to focus attention only on the changed regions. The next sections will describe how object models are formed from the changed regions, and how these models are used to achieve an object persistence system.

B. Multi-view 3D Object Models

Our system attempts to maintain a list of labeled objects that are present in an environment. A human user names each object once and then the system must recognize that object as it moves into new locations and is seen from different vantage points. We describe objects based on their geometric appearance. In particular, we form VFH descriptors [8] for each view of each object. Computing VFH requires a region-of-interest operator, and for this we use the results of the change detection previously described. Each view v_i of each object is then represented as a 308 dimensional feature vector $f_i = f(v_i)$ which describes the 3D appearance. Object recognition occurs by matching the features extracted from changed regions in new views ($f_n = f(v_n)$) to the entire set of vectors previously extracted within the set of known persistent objects. The nearest neighbour search, $argmin_i d(f_i, f_j)$ is performed efficiently using the Floating Point Approximate Nearest Neighbours (FLANN) package provided by Muja *et al.* [14]. With this approach we can practically compare changed regions to many views of many known object models in real time.

There are several outcomes from the nearest neighbour matching. First, the changed region might unambiguously match to one of the persistent object features that have already been learned by the system. In this case, the recognition system returns a confident detection result for later processing by the top-level persistence component. Another common case is that the changed region matches with a large distance to a number of learned features. We apply a threshold to the nearest neighbour distance and the recognition module returns the result that the changed region is likely from a semantic category that has previously not been seen by the system. This region can be passed to the user in a query to obtain a new label.

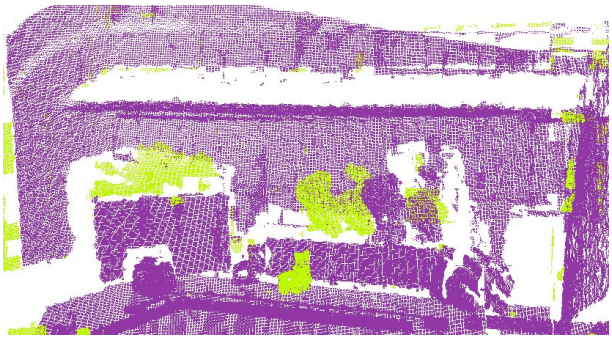
Our object model also allows each object's appearance to be expanded over time. This is an important feature because the system often observes different viewpoints of an object at each time, and the features extracted from each of these viewpoints may not be identical (i.e. in the case of non-symmetric objects). By merging the feature vectors observed over time, the system continues to increase its confidence in the representation of each object, which can provide improvements in accuracy as time progresses.

C. Top-Level Object Persistence Model

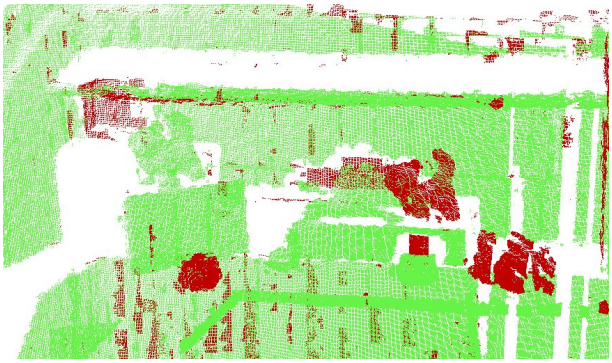
The previous sections describe a method to locate interesting regions and to match these regions to known named objects that are being tracked within the persistence system. However, there are a number of outcomes from each of these modules, and we combine their outputs with a final top-level reasoning procedure in order to maintain the system's state of the objects over time. When our system enters a new environment, it is unaware of the presence of any objects. It initially scans the world and simply memorizes the current 3D geometry. Object discovery for our system begins when objects begin to move. This triggers the change detection system, and initially each changed object is marked as *new* since the database of known objects is empty. The detection of *new* objects results in the human user being queried for a label: "What is this object?". The user may choose to ignore the object, in which case no model is learned, or a name may be entered: "This is a dragon". Once an initial set of objects is known by the system, further visits to the same scene will continue to result in newly moved objects, and there are now several outcomes possible:

1) The location where an object previously existed is now unoccupied.
2) A previously unoccupied region now contains a known object.
3) A location which previously contained one object now has slightly different geometry sensed, and this matches well to a different object.
4) Any type of change is detected where the new local signature cannot be found in the existing database.

Our top-level reasoning system assigns the following outcomes to the cases above: 1) Indicates that a known object was removed from the space. Notice that, in this case, a part of the scene background that was unobserved before now be uncovered. Presently, this can lead to a small number of unnecessary queries to the user for object labels, but we leave sophisticated processing of this case for future work. Cases 2) and 3) indicate that one or more known objects have been moved, in which case the object persistence database is updated, and the new location for each object is now utilized for tasks. The feature vector extracted from the new view is merged with the previous appearance model as has been described above. Case 4) indicates that a previously un-modeled object has appeared in the scene. The reasoning system prompts the user for a label and adds this information to the persistence database.



(a) Detected changes between first and second day



(b) Detected changes between second and third day

Fig. 2. The results of our point cloud registration and differencing module allow the system to focus attention on moved objects.

This completes the description of our system's components. We will continue by describing a set of experiments that have been conducted to evaluate the ability of our approach to maintain object persistency as the environment changes over time.

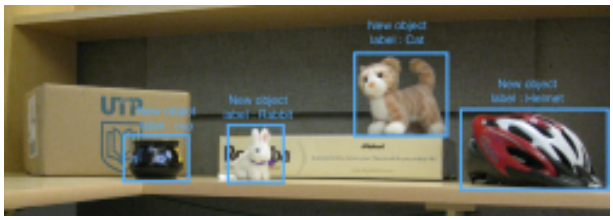
IV. EVALUATION

We have evaluated each component of our system on a variety of environments that contain a number of moving objects. The Microsoft Kinect sensor was used to collect a variety of views of each scene on each *day* and our system was run, querying the user as needed and updating its model when moved objects were detected. This section will describe the performance of each of the model components, and provide numerical analysis comparing the object locations hypothesized by our method at each time against object locations annotated by a human supervisor.

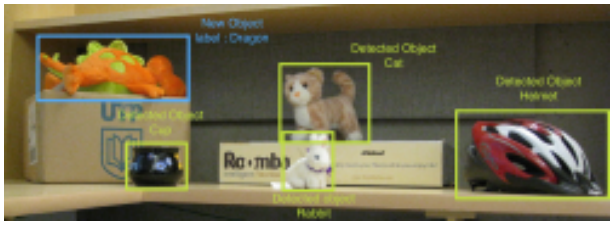
A. Evaluation of the Attention Operator

We begin by examining the results of our registration and change detection module: the attention operator. In each of the environments considered, our ICP registration approach was able to solve for the transformation between viewpoints with sufficient accuracy for the point clouds to be placed in a common coordinate frame. Figure 2 provides an example result of the differencing operator that was subsequently applied. Highlighted changes can be seen to correspond both to the objects that have moved, regions of the point cloud that do not overlap between the views, and small elements of sensor noise. Our algorithm filters the small regions by enforcing a minimum object size. We found that the change detection module was able to correctly locate object regions in areas of moderate to heavy clutter, but in extreme clutter (many objects piled upon eachother), change regions began to differ from the ideal segmentation of objects. We leave dealing with this situations for future work, and plan to investigate global segmentation and object matching approaches similar to [1] in the future to address this problem.

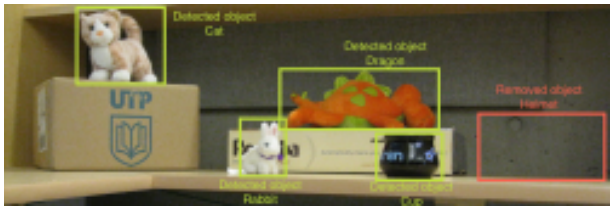
B. Evaluation of Object Recognition and Persistency



(a) User labeled regions



(b) First day hypotheses



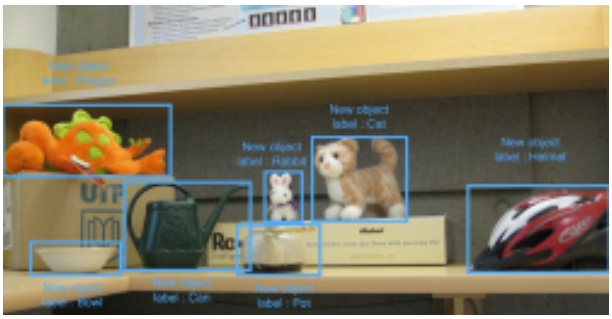
(c) Second day hypotheses

Fig. 3. Qualitative object persistence results over a period of three days for a five object environment. Best viewed in colour and the scheme used is blue for new objects, green for correct re-detections and red for detection errors.

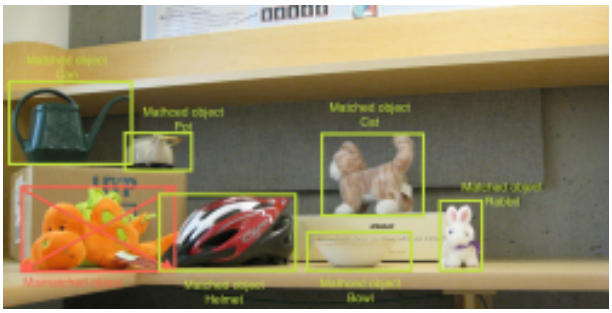
We have tested the core recognition and persistency components of our system by repeating the process of repeatedly visiting an environment where objects are moving, querying the user for initial labels and updating the models as changes occur. We report on these results both qualitatively and quantitatively in the remainder of this section.

Qualitative Results

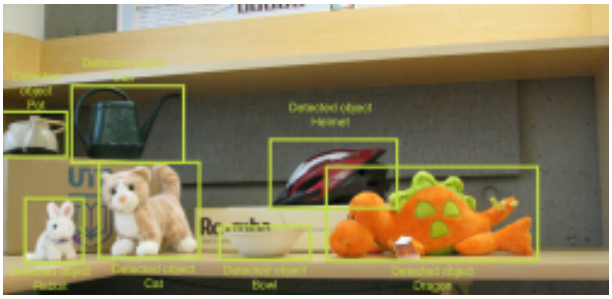
Figure 3 provides one example sequence of the interactions that our system performs as it encounters a set of changing objects in an environment. The first image is captured after a set of objects have been placed into a previously empty scene. The robot detects the changed regions and notes that it has not previously built models for these objects (added objects shown in blue). The user is queried for labels and object models are added to the



(a) User labeled regions



(b) First day hypotheses



(c) Second day hypotheses

Fig. 4. Qualitative object persistence results over a period of three days for a seven object environment. Best viewed in colour and the scheme used is blue for new objects, green for correct re-detections and red for detection errors.

peristent database. The second image shows the state of the world when the robot re-visits the scene the next day. The toy cat and bunny have been moved to new positions and another object is added. The system correctly tracks all moved objects and queries the user for a new label for the object now identified as "dragon". The third image shows the system's state after another visit is made, and the object layout is again changed. In this case several changes are correctly tracked, and a bike helmet is identified as being moved from the scene. Note that we have chosen this scene to demonstrate the useful output provided when we achieve perfect system performance, as no errors were made by our system in identifying the objects and changes in this particular scene.

Figure shows a second qualitative result that uses the same colouring scheme for correct system decisions. However, in this case we must note that in the second image both the dragon and cat are assigned the label "cat" by our system. This is an allowed outcome because the system can handle multiple objects of the same type, but in this case it is in fact an error. We will note in the empirical results and in a later scene that our system does occassionally confuse objects, but we are glad to report this has been rare in our tests.

Quantitative Results

We have also emperically evaluated the performance of our system by comparing the system decisions made at every time with human-labeled ground truth object layouts. The scenario for numerical evaluation is identical to that depicted in the previous qualitative results. That is, an empty scene is observed, a set of objects are initially inserted and the system must query the user for corresponding labels. Then, the objects are moved in between visits made by the robot. The system outputs its best current hypothesis about the object locations at each time, and these hypotheses are compared to ground truth labels created by a human annotator (evaluation independent of any user-interface interaction with the system itself). We use the Pascal Visual Object Categories (VOC) [15] scoring criteria to evaluate correct and missed detections, as this is the current standard in the recognition community. Specifically, the intersection of hypothesized object bounding box is divided by the union of the boxes and this ratio must be greater than 0.5 for correct detections. Tables I through IV display quantitative results where the each cell indicates the number of captured views of an object whose true label is shown in the row which match most closely to the hypothesized label shown in the column. Our scheme for making final decisions is to take a majority vote over these matches. To ease analysis, correct votes (i.e. those along the diagonal) are marked in green and any incorrect vote is highlighted in red. For reference, we note that Tables III and IV are the numerical results that correspond to the scene with seven objects shown in Figure 4. Images for the numerical results in Tables I and II have not been shown, but the scene exhibits similar properties.

In each of the two environments used for scoring, our system performs reasonably well on the first time that objects are moved and then improves to nearly perfect performance on the second round of object changes. We note that the addition of new feature vectors for each correctly identified object that occurs in between visits does lead to a more complete representation of the 3D appearance of each object, and therefore we expect performance to improve over time. We do not claim that the system will continue to perform perfectly for the remainder of time, but this result does certainly indicate the approach is able to maintain a powerful object representation in environments such as those shown here.

V. CONCLUSIONS

This paper has presented a method for object persistence – the task of representing all of the objects that have previously been seen and described to the robot and of updating this representation when changes occur. Our system is based on 3D change detection and descriptive multi-viewpoint feature vectors that allow precise re-recognition

TABLE I

RECOGNITION RESULTS ON THE FIRST DAY FOR SCENE 1

	Cat	Car	Pot	Dragon
Cat	7	0	0	0
Car	0	1	6	0
Pot	6	0	1	0
Dragon	0	1	0	6

TABLE II

RECOGNITION RESULTS ON THE SECOND DAY FOR SCENE 1

	Cat	Car	Pot	Dragon
Cat	7	0	0	0
Car	0	5	2	0
Pot	0	0	7	0
Dragon	0	0	7	0

once objects have been labeled. We allow for objects to be moved within a scene, for new objects to be added and for objects to be removed, gracefully updating the model in each case. Our experiments demonstrate the feasibility of our suggested approach on several somewhat cluttered home-like environments. The system was able to maintain a consistent representation of the locations of objects over the course of several days where the objects were changed between each day.

We believe object persistence, life-long mapping, unsupervised object discovery and object-based mapping are all crucial components for scaling up the semantic understanding of current robots to entire home or hospital-sized environments. It is prohibitively expensive to run object recognizers for the wide variety of object categories of interest continuously and over every portion of a robot's sensory experience. Therefore, it is critical to guide the robot's attention and to have an approach to re-use information whenever possible, as our system exemplifies. However, we have not solved all problems in perceptual memory. Our system relies on a greedy clustering step of the changes detected. This is effective even in somewhat cluttered areas, but breaks down when many objects are moved together, especially when two objects are always touching, such as a cup and saucer. We plan to continue our work on region segmentation.

TABLE III

RECOGNITION RESULTS ON THE FIRST DAY FOR SCENE 2

	Cat	Helmet	Pot	Dragon	Rabbit	Bowl	Can
Cat	4	0	0	0	0	0	1
Helmet	0	5	0	0	0	0	0
Pot	0	0	5	0	0	0	0
Dragon	5	0	0	0	0	0	0
Rabbit	0	0	0	0	5	0	0
Bowl	0	0	0	0	0	5	0
Can	0	0	0	0	0	0	5

TABLE IV

RECOGNITION RESULTS ON THE SECOND DAY FOR SCENE 2

	Cat	Helmet	Pot	Dragon	Rabbit	Bowl	Can
Cat	5	0	0	0	0	0	0
Helmet	0	5	0	0	0	0	0
Pot	0	0	5	0	0	0	0
Dragon	0	0	0	5	0	0	0
Rabbit	0	0	0	0	5	0	0
Bowl	0	0	0	0	0	5	0
Can	0	0	0	0	0	0	5

REFERENCES

[1] E. Herbst, X. Ren, and D. Fox, "Rgb-d object discovery via multi-scene analysis," in *In proceedings of the IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS)*, 2011.

[2] R. Triebel, J. Shin, and R. Siegwart, "Segmentation and unsupervised part-based discovery of repetitive objects," in *In proceedings of Robotics Science and Systems (RSS)*, 2009.

[3] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in *In proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 2168 – 2173.

[4] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, D. G. Lowe, and B. Dow, "Curious george: An attentive semantic robot," *Robotics and Autonomous Systems Journal Special Issue on From Sensors to Human Spatial Concepts*, vol. 56(6), pp. 503–511, 2008.

[5] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *In proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 1156 – 1163.

[6] N. Hawes, M. Hanheide, J. Hargreaves, B. Page, H. Zender, and P. Jensfelt, "Home alone: Autonomous extension and correction of spatial representations," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA 2011)*. Shanghai, China: IEEE, May 2011.

[7] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions On Pattern Analysis And Machine Intelligence (PAMI)*, vol. 21, no. 5, 1999.

[8] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *In proceeding of the IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS)*, 2010.

[9] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *The IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 05/2009 2009. [Online]. Available: http://files.rbrusu.com/publications/Rusu09ICRA.pdf

[10] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard,

"Point feature extraction on 3D range scans taking into account object boundaries," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[11] M. Muja, R. B. Rusu, G. Bradski, and D. Lowe, "Rein - a fast, robust, scalable recognition infrastructure," in *ICRA*, Shanghai, China, 09/2011 2011.

[12] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *In proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1991.

[13] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universitaet Muechen, Germany, October 2009.

[14] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *In proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, 2009.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html, 2010.