World Modeling for Tabletop Object Construction

Arda Inceoglu, Melodi Deniz Ozturk, Mustafa Ersen, and Sanem Sariel
{inceoglua,ozturkm,ersenm,sariel}@itu.edu.tr

Artificial Intelligence and Robotics Laboratory
Istanbul Technical University, Istanbul, Turkey

Abstract. In tabletop construction scenarios, robots work with vertically or horizontally stacked object structures. In order to form such structures, they need to recognize and correctly model closely placed objects in such structures. Depending on the robot's point of view and the objects' positions, it is likely that objects closely located or in contact partially occlude each other, and as a result it is not always possible to model object stacks by relying only on object recognition. However, if the objects are added to the construction consecutively, it becomes possible to sequentially build the model of object stacks. In this work, we propose a scene interpretation system to build and maintain a consistent world model for tabletop construction scenarios. To overcome the challenge of modeling object stacks, we extend our previous scene interpretation system with a semi-closed world assumption and by preserving the models of objects in the formed structures even when they are out of sight. Our extension includes the use of spatial object relations, as well as depth-based segmentation results to model not only single objects, but object combinations. In our system, the LINE-MOD algorithm and an enhanced version with HS histograms are used for recognizing objects along with depth-based segmentation for detecting novel objects. We run numerous construction scenarios using building blocks and show that our system can be successfully used for modeling constructed objects.

Keywords. Scene interpretation, Tabletop object construction, Object manipulation, World modeling for tabletop manipulation

Introduction

In order to achieve given goals, robots often need to interact with various objects. For successful interaction, before anything else, they need to collect correct information about the objects in their environment. The required data includes the accurate properties of objects, such as their size, shape and color, their locations in the world and necessary inter-object relations. For this purpose, robots use their sensors to gather observations from the world, which sometimes do not overlap, are not complete, and sometimes even contradict with each other. Our previous work presents a scene interpretation system to cope with these challenges for a ground robot [1], [2]. In this work, we focus on tabletop object construction scenarios and extend our previous work for modeling stacked objects during task execution. This is mainly important for continually monitoring execution against anomalies (e.g., effects of external interventions) or unexpected

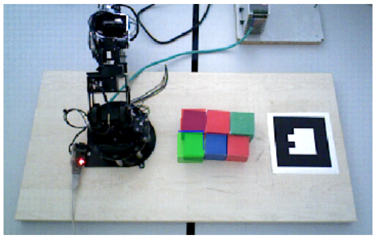


Fig. 1. A rectangular structure is to be built from six cubical blocks by a robot arm. The blocks are from different colors and sizes. When all these blocks are attached to each other, it may not be possible to recognize all of them at once.

outcomes (e.g., inherent failures). New objects should be correctly localized with their properties interpreted and information on these objects should be maintained against any changes (e.g., after disappearing from the scene or displacement in any way). Besides, when symbol grounding is needed for further cognitive skills such as reasoning and learning, correct identification of objects is a prerequisite.

World modeling is especially challenging when objects are in direct contact with each other either horizontally or vertically (i.e., when they are on top of each other). In these scenarios, it is likely that they partially occlude one another from the robot's point of view or the vision algorithms may fail in recognizing all objects. For example, consider the scenario where the robot is tasked to build a rectangular prism structure from a set of cubical blocks in different colors and sizes as in Figure 1. In this figure, the LINE-MOD algorithm [3] is used to recognize textureless objects in 3D by considering their surface normals and matching existing measurements with their previously registered templates. However, since the objects are attached to each other, their boundaries and some of their surfaces can not be distinguished well which results in errors in recognition of some of the blocks (e.g., only two blocks on the leftmost column, indicated with the corresponding markers in the figure, are recognized for this scenario). This problem can be alleviated by reducing the similarity threshold used for matching templates in the algorithm. However, this may result in false positives. Humans, on the other hand, intrinsically use their background and default knowledge when faced with similar problems, incorparating the recent history of events that have led to the current situation. In this study, we are inspired by this cognitive skill and propose a system to reach logical conclusions, similarly to humans, about the robots environment.

Given the requirements in modeling objects in construction scenarios, we propose new advancements over our previous scene interpretation system. The contributions of this work are three fold. First, the scene interpretation system is made capable of using observations taken during execution and building the model of a structure incrementally using both temporal and spatial relations extracted during runtime and prior semantic rules for handling occlusions. Second, a truth maintenance mechanism is applied to store the models of occluded objects even if they can not be recognized but to remove their models when they are believed to disappear from the scene. Third, for the identification of objects a semi closed-world assumption is applied for symbol grounding.

The rest of the paper is organized as follows. First, we mention studies related to world modeling. Then, we describe our scene interpretation system for consistent world modeling in tabletop block construction scenarios. We then give empirical results of our system followed by the conclusions.

Related Work

Several recent studies address the issue of maintaining a world model from the robot's visual observations. Nyga, Balint-Benczedi and Beetz (2014) proposed an ensembles of experts approach based on Markov Logic Networks [4] for fusing different aspects of information coming from different object recognition methods (e.g., LINE-MOD [3], Google Goggles etc.) enabling robots to answer logical queries about different aspects of recognized objects [5]. WIRE [6] is a system based on multiple hypothesis anchoring for robots to maintain semantically rich world models in unstructured and dynamically changing environments. It relies on multiple model tracking for incorporating prior knowledge and multiple hypothesis tracking-based data association for consistently updating the world model using new observations. Another similar study addresses the data association problem from a different perspective by using clustering-based approaches instead of multiple hypothesis tracking [7]. In our previous work, we presented a temporal scene interpretation system for maintaining a consistent world model relying on noisy perception outcomes [1]. Our system uses segmentation outcomes as well as object recognition outcomes to be able to detect objects without previously generated recognition models as unknown object candidates, updates the world model by evaluating these perceptual outcomes temporally, and takes the robot's field of view into account during these updates. In this paper, we enhance our scene interpretation system in the following directions. First, we replace the 2D model of the robot's field of view we used for our ground robot with a 3D model which is necessary for tabletop object manipulation scenarios. Second, we incorporate a semi-closed world assumption for keeping track of previously encountered objects. Finally, we present enhancements for the block construction domain.

Perception Sources

The first step in object manipulation by autonomous robots is maintaining a consistent and up-to-date world model about their environment. For this task, the robot has to collect visual recognition and detection data to filter out and to reach conclusions. Our perception system uses LINE-MOD [3], LINE-MOD&HS [8] and 3D segmentation [9] algorithms as means for processing 3D sensory data obtained from an on-board ASUS Xtion Pro RGB-D camera. LINE-MOD is an object recognition algorithm that uses surface normals of the objects, calculated from the Point Cloud [10] data regarding the object, to extract object templates. The algorithm then uses these templates with the sliding windows approach to detect the modelled objects in new scenes. The LINE-MOD&HS algorithm, in turn, augments LINE-MOD to use the HSV histograms of the objects in order to integrate the use of color information of the objects in recognition.

Additionally, 3D segmentation is used for detecting objects that are either not previously modelled, or otherwise cannot be recognized in the current scene.

The perception sources are implemented as separate processes, where their recognition/detection results are asynchronous. The Scene Interpreter system combines these results to create an accurate representation of the world [1].

Scene Interpretation for Tabletop Manipulation

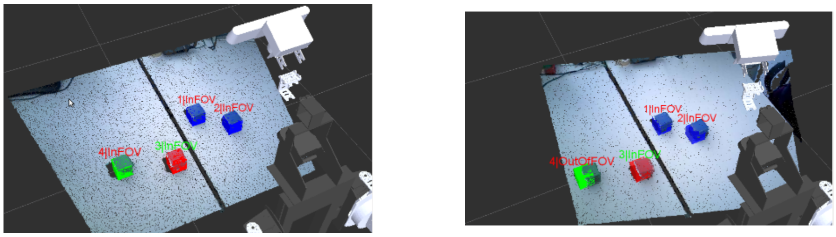
Object recognition is not reliable alone for robotic manipulation tasks, since failures in recognition or detection occur due to noisy sensor measurements, illumination changes, dynamic environments or other agents and sensors. In order to automatically build a consistent and up-to-date model about the environment, visual recognition and detection outcomes should be filtered out and logical conclusions should be reached in the face of contradictory outputs. Previous work by the same research team includes a Scene Interpretation system for ground robots working with objects clearly separated in the horizontal plane [1], which forms the foundation of the proposed system. Necessary deductions about a robot's environment include a unique id for each object in the environment, their type, color, size, shape and location properties, as well as the confidence of the system about these object's existence in the environment.

The confidence is represented with a value varying between 0 and 100, proportional to the degree of belief on the corresponding object's existence. Confidence values are updated with every new perception outcome. An object's confidence value increases as more consistent recognition or detection results arrive regarding the object.

The observed facts are kept in the Knowledge base (KB) of the robot, which can be defined as a collection of reached conclusions about objects, their properties, and inter-object relations. The robot's KB is initialized as empty. During run time, recognized objects are inserted into the KB and their corresponding confidence values, as well as properties, are updated with each newly received recognition message. If an object in the KB does not receive any corresponding recognition message for a period of time, even though this object is in the robot's field of view and should be recognized, the confidence value regarding the object is gradually decreased. If this value reaches zero, it is believed that the object is no longer in the robot's environment, and thus it is removed from the KB.

Most humanoid robots have the capability of moving their heads around, making it possible for them to observe more about their environment. As a result, their visual field of view (FOV) is bounded by the limitations of their cameras. A robot can receive reliable information about objects only within its FOV and the field of view constraints should be taken into account when updating object properties. Extending the 2D definition in the base system, 3D boundaries are empirically determined for an RGB-D camera where the objects within are expected to be recognized reliably. An example scenario regarding FOV calculations can be seen in Figure 2.

Objects in the environment are considered depending on whether they are inside the camera's FOV or not. Objects outside the FOV are not expected to be recognized, and any data regarding them in the KB are kept static until they re-enter the FOV of the robot, and new perception data are available.



(a) (b)

Fig. 2. A scenario demonstrating FOV calculations. In (a) all objects are in FOV. In (b) the robot's head is slightly rotated to right. This time the green block becomes out of FOV, yet it is kept in the KB. (Note that FOV area is determined tighter than the actual physical limits of the camera for more reliable object recognition, and it can be adjusted easily.)

Spatial Relations

After recognition and localization of the objects in the scene, their spatial relations are determined as in [1]. These relations are represented as unary or binary predicates such as $onTable(obj1)$, $near(obj1, obj2)$ and $on(obj2, obj1)$. Consider a scenario where three blocks are stacked on top each other, assigned ids 1 to 3 from bottom to top. There are two on relations expected such that $on(3, 2)$ and $on(2, 1)$. Objects in the bottom are considered as out of field of view and thus they are not expected to be detected. We make use of this for modeling objects in block construction as visualized in Figure 3.

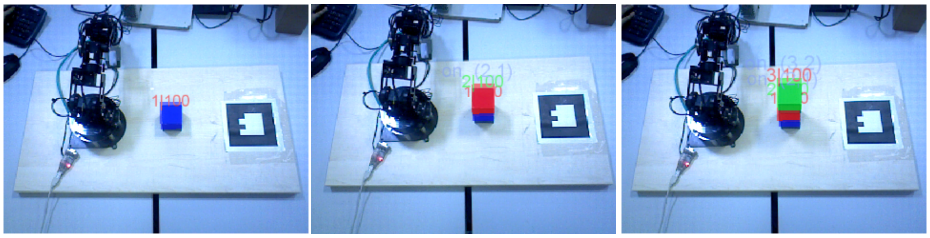


Fig. 3. The phases of a block stacking scenario in the real world (from left to right). The recognized objects along with their ids and confidence values, and the relations among them (e.g.,$on(obj_2, obj_1)$) are marked on the original RGB image in Rviz.

Symbolic Models of Tabletop Objects

The first part of the study focuses on symbol grounding problem for objects ids. Ambiguities in determination of ids can arise in dynamic scenes. Objects may be displaced, removed from and put back into the scene, or the robot could be mobile and have localization problems. As a result, an object might be registered with different ids over time, which prevents creating and executing plans including object manipulation successfully.

After successfully detecting objects, the world is assumed to be closed (closed world assumption) for identity resolution tasks. Closed world assumption [11] can be defined as having complete knowledge about the world, that is, the numbers and the attributes of all objects are known apriori. However, robots often have partial information about the world. Even though object attributes are known, objects' locations may be dynamic or unknown which requires obtaining extra information from the environment [12]. Whereas, in an open world assumption, no prior knowledge about the world is given, and every object entering to the scene is assumed to be encountered for the first time. In contrast, we define a semi-closed world assumption in which the robot builds its KB itself at runtime and does not use any prior knowledge about the scene contents. At each object detection, the attributes of the object are compared with that of previously registered objects in the KB. If an object is believed to have been encountered before, its previous id is used, otherwise a new id is generated. The corresponding algorithm is given in Algorithm 1.

```
Data: Detected object attributes
Result: Object Id
foreach $object$ in $KB$ do
    if attributes match and object is not in the scene then
        return $object.id$;
    else
        $newId \longleftarrow$ generate new id ;
        return $newId$
    end
end
```

Algorithm 1: Algorithm for Semi-Closed World Assumption

Symbolic Relations among Tabletop Objects

The second focus of the study is to correctly model closely located objects. Object recognition in cluttered scenes is still a challenging problem. Object detection success is low in such scenes due to their placements. An example scenario with six cubical blocks is given in Figure 4. The system can not distinguish between objects and only some of the objects can be added to KB. This is the natural result of assumptions of vision algorithms. LINE-MOD extracts surface normals on visible surfaces and color gradients around borders. Furthermore, the 3D segmentation algorithm assumes objects are clearly separable on a supporting plane.

The first solution attempt to this problem was decreasing the similarity threshold of the LINE-MOD algorithm between the object templates and real time detections (See Figure 5). The threshold is set to 95% by default, and it is decreased gradually. As a result, the system was able to detect the objects and register them into the KB. The main drawback of the approach is, as the threshold is lowered, the number of false positives, i.e. the number of misdetections increase. As the threshold reaches 80% and below it becomes harder to maintain the number of objects in the KB.

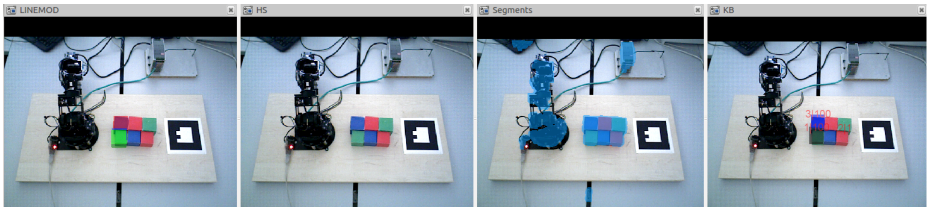
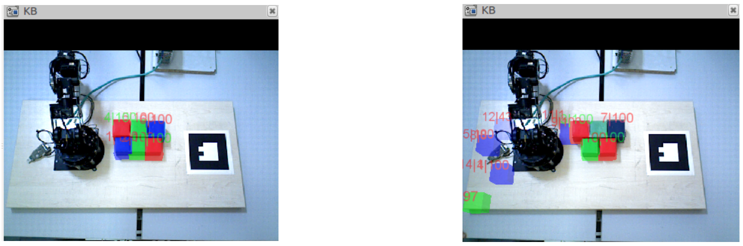


Fig. 4. The first three frames represent the outputs of algorithms LINEMOD, LINEMOD&HS and Segmentation, respectively. Rightmost frame represents the state of the KB, where recognized objects are marked with their ids and a confidence values. The six cubes are placed into the scene initially, only 3 of them are recognized and added to KB.



(a) (b)

Fig. 5. Scenarios with different similarity thresholds for the object recognition algorithm. Thresholds are set to 90% in (a) and 85% in (b). Comparing with Figure 1, in (a) objects are recognized and added to the KB. For the case of (b) false positives are introduced by decreasing the threshold.

For a similar scenario where the threshold is set to 95%, even though objects are placed into the scene one by one but very closely, some of the previously recognized objects are removed from the KB due to lack of recognition messages after some point. The second proposed approach utilizes the 3D segmentation algorithm. We can rely on detections of the 3D segmentation algorithm in terms of the existence of an object in the scene. If the objects are placed into the scene one by one and clear enough to be recognized, after successfully being added to the KB, the objects can be marked as detected, if their centroid lies in one of the last segmented point clouds produced by the 3D segmentation algorithm. Thus, the objects are exempted from being removed from the KB. The proposed algorithm is given in Algorithm 2.

The segmentation algorithm is used to maintain object stacks of the same level. In order to increase the level -the height- of the structure, spatial relations among objects, namely *on* relations, are employed. When objects are stacked on top of each other, corresponding *on* relations are detected between object pairs. In a pair, the bottommost object is partially occluded, so it is not expected to be detected, which avoids the update operation on the object and thus the deletion from the KB. In addition, if the topmost object of a pair is removed from the scene, the corresponding object model and the *on* relation are also removed from the KB, and the bottommost object is expected to be detected again.

```
$KB \longleftarrow$ initialize empty knowledge base;
upon receive $Objects$:;
/* $Objects$ : Recognized objects via LINEMOD and LINEMOD&HS */
foreach $object$ in $Objects$ do
    if $object$ in KB then
        update $object$;
    else
        $KB \longleftarrow$ add $object$ ;
    end
end
upon receive $Segments$:;
/* $Segments$ : Segmented point cloud clusters */
foreach $object$ in $Objects$ do
    if $objectcentroid$ in $Segments$ then
        $object \longleftarrow$ mark object as detected
    end
end
```

Algorithm 2: Maintaining Closely Located Objects

Experiments

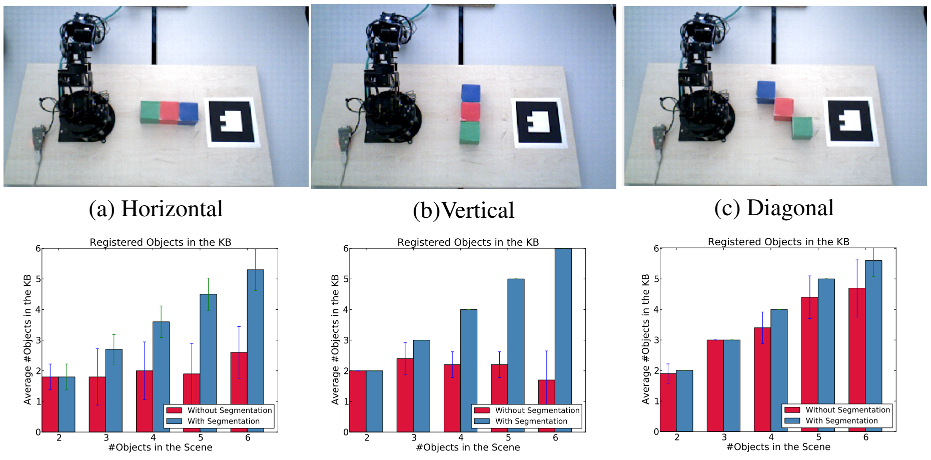
This section describes the experimental setup and presents the obtained results. First, we present object recognition and registration to KB during run time. Then, id tracking capabilities of our system under semi-closed world assumption is demonstrated.

Object Registration to the Knowledge Base

For the first part, block construction scenario is considered. Red, green and blue colored blocks are placed into the scene sequentially to form horizontal, vertical and diagonal structures on the same plane. For comparison purposes, experiments are repeated with and without employing the proposed segmentation based approach. Each time, after a block is placed, the number of objects registered to the KB is recorded. Each case is repeated 10 times, and the mean is calculated. Figure 6 shows the comparison for horizontal, vertical and diagonal structures. Note that, since the results are recorded in a sequential manner, errors in the previous steps accumulated to oncoming steps.

The following conclusions can be drawn from the analysis given in Figure 6. Employing segmentation based approach fairly increases the number of objects registered to the KB. The best performance is obtained from vertical placement scenario due to the fact that the last placed object can be correctly isolated from its surroundings and thus, it is easier for the vision algorithm to recognize. However, in the diagonal placement scenario using segmentation does not provide much improvement since objects are in less contact with each other. Horizontal scenario is the most complicated one in terms of distinguishing between objects, since objects have more contact with each other. Improvements become clear when the number of objects in the structure is increased.

In another experiment, blocks are stacked on top of each other one by one to measure *on* relation detection success when new layers are introduced. This time, after each



(a) Horizontal (b)Vertical (c) Diagonal

(d) Horizontal (e)Vertical (f) Diagonal

Fig. 6. Comparison of the number of registered objects into the KB.

stacking operation, the number of *on* relations is recorded. It is expected to detect one *on* relation with a stack of two objects, two *on* relations with a stack of 3 objects and so on. Success rates of detecting *on* relations are 100%, 100% and 93.3% for the number of layers 2,3 and 4 respectively. Due to object recognition failures, success rate is decreased in the 4th layer and above.

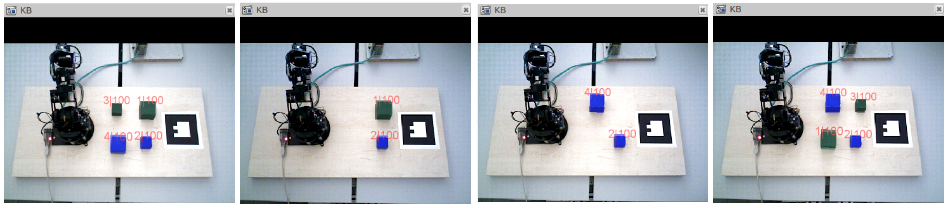
Semi-Closed World Assumption

The goal of this experiment is to illustrate id tracking capabilities of the system. The object set contains red, green and blue colored, medium sized cylinders, and small and large sized blocks. In the model; type, size and color attributes are taken into account. An example scenario is visualized in Figure 7.

The KB is initialized by putting all target objects into the scene, and each object is assigned a unique id. Then, the objects are removed from the scene. Each object is put back and id assignments are observed. A confusion matrix based on id assignments is given in Figure 8. Whenever an object could not be matched with the previously encountered objects registered to KB, a new id is generated for the object. The reason of mismatches are originated from errors in recognizing objects due to illumination conditions. It is observed that if an object is failed to match with an object in the initial object set, and thus attached a new id, the consecutive recognitions are also matched to this id. Whereas, some objects could not be recognized at all, which are denoted as not detected in Figure 8.

Conclusion

We have presented enhancements for our scene interpretation system in order for it to be used in tabletop manipulation and construction scenarios for cognitive robots. First,



(a) (b) (c) (d)

Fig. 7. A scenario demonstrating the id tracking performance of the system.(a) contains small and large sized blue and green cubes. Object are added to the KB and each is attached a unique id. In (b) the small green cube and the large blue cube are removed. In (c) the large green cube is removed and then the large blue cube is placed again. In (d) all objects are put back into the scene. By using the size and color attributes, the system is able to remember the objects and attach their previous ids.

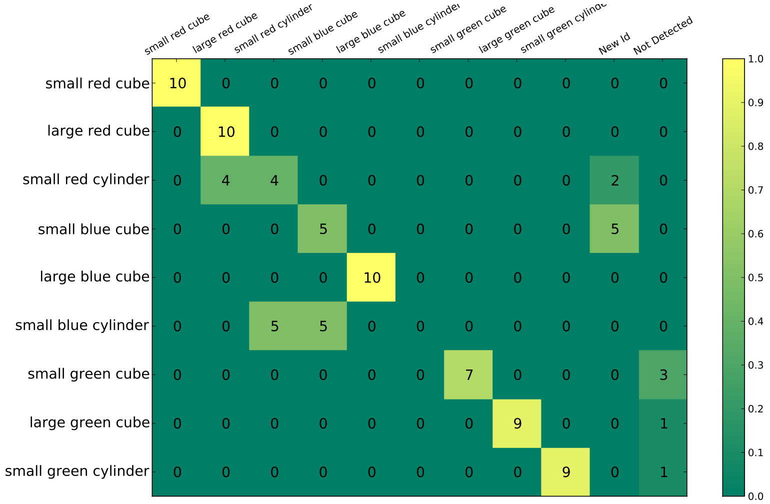


Fig. 8. Confusion matrix for semi-closed world assumption. After each object is assigned a unique id, objects are removed and put back. Matrix shows original versus newly assigned ids. A new id is assigned if object is recognized yet could not be matched with previous objects. Not detected denotes, object could not be recognized at all.

the 2D model of a ground robot's field of view was extended to 3D for a humanoid robot with a moveable head. Then, we introduced a hybrid model of open and closed world assumptions for keeping track of object ids in case of dislocations and disappearances & reappearances. This hybrid model is able to keep track of lost objects, while still allowing new objects to enter the scene. Deductions about object ids are made based on physical attributes of the objects and without using any kind of prior knowledge. Finally, for block construction scenarios, we proposed utilizing 3D segmentation on top of object recognition to maintain objects in the KB when they are in direct contact with each other and cannot be recognized. Furthermore, we employed spatial relations to maintain objects that have other objects on top of them and thus to model higher level structures. Future work includes improving the system to keep track of more complicated scenarios that include unknown objects, and modifying the system to operate on a probabilistic framework.

Acknowledgments

This research is partly funded by a grant from the Scientific and Technological Research Council of Turkey (TUBITAK), Grant No. 111E-286.

References

1. M. D. Ozturk, M. Ersen, M. Kapotoglu, C. Koc, S. Sariel-Talay, and H. Yalcin, "Scene interpretation for self-aware cognitive robots," in *Proceedings of the 2014 AAAI Spring Symposium: Qualitative Representations for Robots*, pp. 89–96, AAAI Press, 2014.
2. M. Ersen, M. D. Ozturk, M. Biberci, S. Sariel, and H. Yalcin, "Scene interpretation for lifelong robot learning," in *The 9th International Workshop on Cognitive Robotics (CogRob 2014) held in conjunction with ECAI-2014*, (Prague, Czech Republic), 2014.
3. S. Hinterstoisser, C. Cagniart, S. Ilic, P. F. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 876–888, 2012.
4. M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.
5. D. Nyga, F. Balint-Benczedi, and M. Beetz, "PR2 looking at things - Ensemble learning for unstructured information processing with markov logic networks," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3916–3923, IEEE Press, 2014.
6. J. Elfring, S. van den Dries, M. J. G. van de Molengraft, and M. Steinbuch, "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, 2013.
7. L. L. S. Wong, L. P. Kaelbling, and T. Lozano-Pérez, "Data association for semantic world modeling from partial views," *International Journal of Robotics Research*, Accepted for publication.
8. M. Ersen, S. Sariel-Talay, and H. Yalcin, "Extracting spatial relations among objects for failure detection," in *Proceedings of the KI 2013 Workshop on Visual and Spatial Cognition*, pp. 13–20, 2013.
9. A. J. B. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," in *Proceedings of the 3rd Workshop on Semantic Perception, Mapping and Exploration (SPME)*, 2013.
10. A. Aldoma, Z. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 DOF pose estimation," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 80–91, 2012.
11. R. Reiter, "Readings in nonmonotonic reasoning," ch. On Closed World Data Bases, pp. 300–310, Morgan Kaufmann Publishers Inc., 1987.
12. O. Etzioni, K. Golden, and D. S. Weld, "Sound and efficient closed-world reasoning for planning," *Artificial Intelligence*, vol. 89, no. 12, pp. 113 – 148, 1997.