18th IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
April 25-27 2018, Torres Vedras, Portugal

Planning Object Informed Search for Robots in Household Environments

Marco A. Gutiérrez[1], Luis J. Manso[1]*, Pedro Núñez[1] and Pablo Bustos[1]

***Abstract*— In the current state-of-the-art, social robots performing non-trivial tasks often spend most of their time finding and modeling objects. In this paper we present the extension of a cognitive architecture that reduces the time and effort a robot needs to retrieve objects in a household scenario. We upgrade our previous Passive Learning Sensor algorithm into a full fledged agent that is part of the CORTEX robotics cognitive architecture. With its planning capabilities, this new configuration allows the robot to efficiently search, pick and deliver different objects from different locations in large households environments. The contribution presented here dynamically extends the robot's knowledge of the world by making use of *memories* from past experiences. Results obtained from several experiments show that, both, the new software agent and the integrated cognitive architecture, constitute an important step towards robot autonomy. The experiments show that the find-and-pick task is greatly accelerated.**

I. INTRODUCTION

When social robots are required to complete basic household tasks, a necessary skill is the ability to explore the environment, find an object, pick it up and move it somewhere else, under different contextual situations. The research community has been very active in the completion of several variants of basic tasks, such as making pancakes [1], or cloth folding [2]. To complete these find, pick and deliver type of tasks, a delicate coordination of several functionalities is required, including navigation, localization, manipulation and grasping, speech understanding, planning and object and human detection and recognition, all of them tied up by some kind of short and long term knowledge representation. Robotics cognitive architectures, like the one used in this work, CORTEX [3], are suitable to achieve the required level of integration, flexibility and adaptive decision making.

When robots have to find objects in large indoor environments, the use of a random search strategy can be too slow for humans and might frustrate potential domestic robot users. In order to speed up this process, robots have to optimize the search process generating hypotheses pointing them to the most-likely object locations [4], [5], [6]. One way of generating these hypotheses is by exploiting previous experiences of the robot as a source of information to weight the set of candidate places for a search task.

The technique presented in the paper extracts the information to infer the potential locations of objects from images acquired by the robot while performing other tasks. We have designed a new element for the CORTEX architecture that passively observes the environment as the robot moves, gathering information about the location of the objects found. Objects are segmented and detected by this passive learning agent using machine learning techniques, and the corresponding information is stored linked to their location in the current representation of the environment. The combination of geometric and semantic attributes is used to infer probable object locations when requested by a human. In particular, we build upon the example of a social robot in a domestic environment that is asked to fetch and deliver different objects placed on several tables. The algorithm exploits the fact that related objects are usually placed together. The tables to inspect are selected based on the semantic distance between the object to find and the description of the types of objects placed on the tables. This helps the robot take decisions about the location of the objects that have not been seen yet, improving its usefulness for the human user.

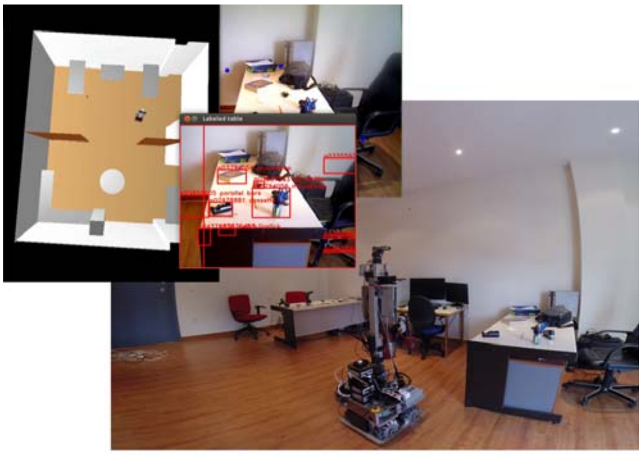

Fig. 1. The robot Shelly "passively learning" while moving around the apartment.

This work has been integrated in a robotic architecture which enables the robot to generate plans and coordinate different software modules in such a way that object discovery can become part of more complex tasks. The main contribution of this paper is the demonstration of how our previous work on multimodal passive learning [7] can be upgraded to a software agent, becoming part of a more complex robotics cognitive architecture -CORTEX- that enables an automated social robot to deliver objects on demand in a large household

*This work has partly been supported by grants 0043_EUROAGE_4_E, from the European Union - Interreg project-, TIN2015-65686-C5-5-R, from the Spanish Government, and by grants GR15120 and IB16090, from the Government of Extremadura.

[1] All the authors are with the Robotics and Artificial Vision Laboratory of the University of Extremadura, Avda. Univesidad s/n, Cáceres, Extremadura, Spain. `lmanso@unex.es`

environment. This integration allows the whole system to respond to human demands, generating plans to fulfill them. In doing so, the robot learns the location of objects and uses this knowledge to improve subsequent searches.

The remainder of the paper is as follows: Section II presents a review of current semantic mapping approaches meant to find objects in indoors environments. Section III introduces the CORTEX architecture, including the multimodal passive learning agent, and explains how the planning of the informed search is conveyed. Section IV presents the tests performed to validate the system, first evaluating the passive learning agent and then using an autonomous robot executing full object search tasks. Finally, Section V presents the conclusions drawn from the research and some future lines of work.

II. RELATED WORKS

Semantic mapping algorithms can be categorized according to the scale in which they work. Regarding those that focus on human indoor environments, a distinction can be done between those focusing on single scenes and those that target large scale ones. The former reason about an instance frame with respect to a local coordinate system while the later progressively annotate a metric map with semantic information, located with respect to a global frame of reference.

In relation to single scene annotation, in [8] a model is presented based on a conditional random field with several constraints to facilitate hierarchical labeling. It also makes use of object class hierarchies to overcome uncertainty when labeling a scene. Relevant objects are extracted in [9], [10] by processing large input datasets and labeling a geometric map. This is done building complete object models from partial 3D object views and feature-based recognition procedures. The objects modeled are kitchen serviceable ones, such as appliances, cupboards or tables, being of specific significance for household assistant robots. In our work, instead of labeling objects we convert this information into location labels that offer information regarding the information at that location, *i.e.*, on a certain table.

Regarding large scale semantic mapping, the work in [6] proposes symbolic representations of qualitative spatial relations. It uses geometric models on these representations with respect to landmark objects. This information is later used to decide which location to search first. Likewise, the works presented in [4], [5] make use of spatial information for large scale scenarios. They formalize the object search as a probabilistic inference problem using different priors. These priors represent several aspects of the scene such as scene structure, physical constrains or domain knowledge. By transforming these priors into a probabilistic model they create hypotheses about possible object locations. These approaches are similar to ours, as they take into account the different information to value locations to approach.

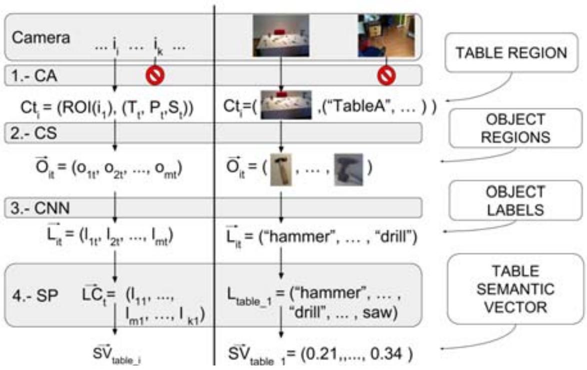
Camera
$\ldots i_i \ldots i_k \ldots$
1.- CA
$Ct_i = (ROI(i_1), (T_t, P_t, S_t))$
Ct_i=(,("TableA", …))
TABLE REGION
2.- CS
$\vec{O}_{it} = (o_{1t}, o_{2t}, \ldots, o_{mt})$
$\vec{O}_{it}$ = (, … ,)
OBJECT REGIONS
3.- CNN
$\vec{L}_{it} = (l_{1t}, l_{2t}, \ldots, l_{mt})$
$\vec{L}_{it}$ = ("hammer", … , "drill")
OBJECT LABELS
4.- SP
$\vec{LC}_t = (l_{11}, \ldots, l_{m1}, \ldots, l_{k1})$
L_{table_1} = ("hammer", … , "drill", … , saw)
TABLE SEMANTIC VECTOR
$\vec{SV}_{table\_i}$
$\vec{SV}_{table\_1}$ = (0.21,,…, 0.34)

Fig. 2. The four main layers of the passive learning agent. The left-hand side of the vertical line describes the output of each layer in a formal notation, while the right-hand side shows it visually. The forbidden sign represents discarded images. Explanations on the outputs are given in the outer right descriptions.

III. PERCEPTIVE PLANNING ARCHITECTURE

The solution presented here uses a combination of large scale and single scene approaches coordinated through the CORTEX architecture. As a first step, a passive fast annotation on the large scale environment is made while the robot moves around the environment. This enables the robot to produce hypotheses on where to look for different objects. When a search task comes, the locations are approached by the robot following these hypotheses. A second stage of single scene semantic mapping is then conducted so that the existence and location of the searched object is verified and integrated in the world model of the robot.

A. Multimodal Passive Learning Agent

The multimodal passive learning agent is a module that is designed to speed up the object search process through the exploitation of robot *memories*. It works by passively processing the images that the robot acquires while moving around the environment, performing different tasks or just waiting for a command. Images are processed through the different layers of the agent to give the robot the means to guess where an object would be most probably located.

The agent implements a processing pipeline of four layers (see Fig. 2. A first layer, Cognitive Attention (CA), selects images that contain a table with potential objects on it. The Cognitive Subtraction (CS) layer, combines the existing model of the world and the incoming image to segment the new elements in it. Afterwards, a Convolutional Neural Network (CNN) layer computes labels for the unknown regions of the image obtained in the previous step. Finally, a Semantic Processing (SP) layer uses a learned semantic model to improve the labels obtained from the CNN and maximizes the probability of finding the correct place for a searched object.

1) Cognitive Attention: This first layer of the agent filters the images that contain a table. Then, it detects the regions of interest (ROI) corresponding to those tables, and provides such regions along the information of the table (identifier and geometrical properties) to the next stage of the pipeline.

For each image I_i accepted, this layer provides the next one with a tuple T_i containing the region of interest of the image that shows the table $ROI(I_i)$, along with table specific information such as its type (T), pose (P), and shape (S). See Eq. 1 for a formal description of this layer.

$$CA(I_i) = T_i = (ROI(I_i), (T, P, S)) \tag{1}$$

In order to estimate if a table lies in the frustum of the camera, the robot uses a model of the environment. In our case, as shown in Fig. 3, CORTEX provides the needed information (the pose of the robot and the tables in the environment) to select the ROIs containing tables (see Section III-B for more details on CORTEX).

2) Cognitive Subtraction: The second layer of the agent performs an additional segmentation step called Cognitive Subtraction (CS). It takes as input the tuples of the regions of interest obtained from the previous layer, along with the table information, T_i, and generates a series of sub-regions of interest out of each image. These sub-regions, o_{jt}, which correspond to object proposals, are associated to their table and constitute the output of this layer. For instance, for one image i, associated with a table t, this layer will produce a set of sub-images $\vec{O_{it}} = (o_{1t}, o_{2t}, ..., o_{mt})$ (see *2.-CS* layer in Fig. 2 and Eq. 2 for a formal description).

$$CS(T_i) = \vec{O_{it}} = (o_{1t}, o_{2t}, ..., o_{mt}) \tag{2}$$

In order to perform this segmentation, this layer uses a simple pipeline that targets objects lying on tables:

1) A random sample consensus [11] is used to estimate the plane of the table using the point cloud of the scene acquired with the RGBD camera. Only points lying over this plane are considered.
2) The remaining point are segmented using euclidean distance clustering [12].
3) Candidate object point are transformed to image coordinates and the image region corresponding to those points is segmented, generating object candidates.

Additional containers can be considered by integrating additional pipelines in this layer.

3) CNN Object Labeling: The third layer classifies the ROIs from the previous step. It produces a label l_{it} for each of the object candidate regions o_{it} obtained from the previous layer (see Eq. 3 for a formal description and *3.-CNN* layer in Fig. 2).

$$CNN(\vec{O_{it}}) = \vec{L_{it}} = (l_{1t}, l_{2t}, ..., l_{mt}) \tag{3}$$

The current implementation uses a very deep Convolutional Neural Network (CNN) based on deep residual learning [13]. Specifically, a generic training of this CNN, with 152 layers, on the ImageNet dataset [14] was used.

4) Semantic Processing: The Semantic Processing step (SP) takes the labels $\vec{L_{it}}$ produced in the previous step for all the images and groups them according to each table t. For each table an average semantic vector is produced using the semantic vector representations of these labels. For each table t and all labels $\vec{L_{it}}$ of each image i:

$$\vec{LT} = \vec{L_{1t}} + \vec{L_{2t}} + ... + \vec{L_{pt}} = (l_{t1}, l_{t2}, ..., l_{tm}, ..., l_{tk}) \tag{4}$$

First, these labels are transformed into each of their 300 dimensions semantic vector representations making use of the skip-gram model [15]. These models are two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word vectors are positioned in the vector space such that words that share common contexts in the training corpus are located in close proximity to one another in the space. A model trained on texts obtained from the Google News dataset (with more than 100 billion words) is used. 300 dimensions are used for the vector representation as it provides good accuracy without dramatically affecting training [16]. Using this vector representations, an average semantic vector $\vec{SV_t}$ is produced for each table t (see step *4.-SP* in Fig 2).

Later on, when the robot is asked to find an object with label l_o, the semantic vector representation $\vec{SV_{l_o}}$ of the label is computed using the learned skip-gram model. Afterwards, the semantic similarity $\mathbb{SS}_t$ to each table t is calculated as the cosine distance (dot product) of the representation of the label $\vec{SV_{l_o}}$ and the semantic vector of the table $\vec{SV_t}$. The higher the value obtained the better result and the closer in the semantic space. The robot will then approach the tables in order of higher semantic similarity with the label of the object searched.

This last step is intended to expand and improve the image labeling results by using the semantic relationships between the labels obtained for a specific table. It makes so by taking advantage of the fact that, in household environments, objects located next to each other are often semantically related (*i.e.*, kitchen utensils are placed together and toys are kept in the same place).

B. Integration of the Multimodal Passive Learning Agent in CORTEX

Here we present the deliberative cognitive architecture, CORTEX, that enables the robot to perform all the steps of the object search. These steps start by asking the passive learning agent for object location hypotheses and guiding the robot through the search, finishing with the verification of the object classification and its precise location in the scene.

CORTEX is built on top of many developments carried out during past years, the most relevant being the *Active Grammar-based Modeling* architecture (AGM) [17], [18], the RoboComp framework [19] or the Deep State Representation concept [20], [18].

1) Agents: Robot's actions and the resulting modifications in the internal model that represents the world are carried out by a set of software modules named *agents*. Depending on the current plan of the robot, each agent will have to execute different actions. In general, the action to be executed depends on the current step of the plan, but it can also depend

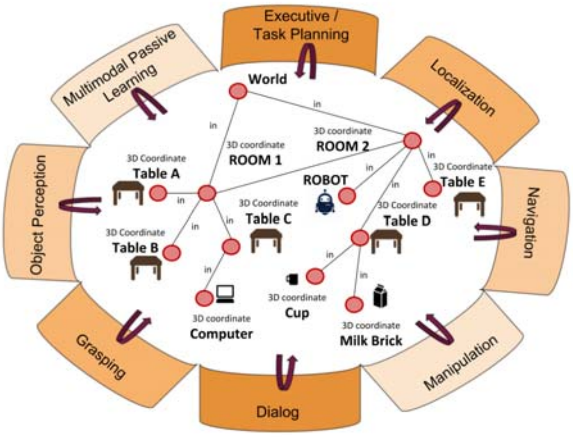

Fig. 3. The CORTEX architecture along with its agents accessing and contributing to the maintenance of a shared representation of the environment and of the robot. Planning and Executive are in charge of high-level planning activities.

on the remaining uninitiated steps[1]. Similarly, each agent can perform modifications in the model, whether these are the result of the action of the robot or an exogenous event[2]. Therefore, depending on the task a robot has to deal with, different agents will be active.

The following agents take part in the implementation of the architecture used for the experiments: a *Navigation* agent in charge of moving the robot; a *Localization* agent in charge of indicating which is the current room where the robot is located at, and its specific coordinates in it; a *Proprioception* agent which continuously updates the angles of all the joints of the robot in the internal model; an *Object* agent in charge of detecting and updating the position of the objects seen in the scene; a *Human* agent which detects and includes the persons nearby into the world model; a *Dialog* agent which updates the mission of the robot upon user request; and the new *Multimodal Passive Learning* agent. Fig. 3 shows the architecture along with the world model and the agents that interact with it.

2) Nodes and Associated Actions: The plan that a robot uses to fulfill its mission depends on its domain knowledge (*i.e.*, the actions that can be performed by the robot), but also on the internal world model of the robot. In CORTEX the world model is a shared hypergraph named Deep State Representation (DSR) –*i.e.*, a graph where pairs of nodes can be linked multiple times. DSRs have the particularity that both, nodes and edges can have any number of additional attributes. These attributes are used to encode metric information which is not taken into account by the planner.

The domain of the robot (*i.e.*, the set of actions that can be executed, along their preconditions and consequences) is expressed using a *grammar* similar to those used to define formal languages. These grammars are sets of grammar rules that are used by the executive to compute the plans to achieve the missions of the robot. Despite there are additional rules designed perceive other objects and rules for other different activities such as human-robot interaction or manipulation, the five following rules have special interest in this work:

- **imagineObjectInPosition** is used to imagine a "proto-object" in the most-likely table so it can later be inspected and confirmed or discarded. It generates a *protoObject* node associated to an already existing table.
- **setObjectReach** is used to get close to objects such as tables or mugs.
- **changeRoom** is used to make the robot change from a room to one of the adjacent ones.
- **verifyImaginaryObject** is used to confirm that a previously imagined "proto-object" has been successfully modeled as an actual object. It basically changes the type of the node from "protoObject" to "object".

IV. EXPERIMENTS

A set of experiments have been made with a real robot in an apartment-like scenario with two rooms. The robot used is an omnidirectional manipulator enabled with a camera, a kinnect sensor and a 2D LRF. Five tables where disposed along the apartment containing five types of objects: table A contains hardware tools, table B has a computer and other tech gadgets, table C has office supplies, table D has kitchen utensils and table E contains different toys. In a first training step the robot wandered around the apartment in order to build the robot "memory", simulating a previous experience. Afterwards, the effectiveness of the multimodal learning agent was tested against other state-of-the-art labeling processes. Final tests where performed with the robot running whole integrated CORTEX architecture.

A. Multimodal Passive Learning Agent Tests

The multimodal passive learning agent was tested against a combination of state-of-the-art segmentation and CNN classification algorithms. The classification algorithms used for the test are the following:

- GoogleNet [21] is a 22 layers deep network (27 if pooling is taken into account) that makes use of "inception modules" which basically act as multiple convolution filter inputs, that are processed on the same source, while pooling at the same time. Another training of this network but without relighting data-augmentation was also tested (GoogleNet2).
- AlexNet, by Krizhevsky *et. al.* [22], consists of eight layers, of which five are convolutional layers, with some of them being followed by maxpooling layers. The other three layers are fully-connected layers with a final 1000-way softmax.
- Very Deep Convolutional Networks by Simonyan *et al.*, presented in [23] (VGG16 in Table I). Consist of a series

[1] Lets assume that a robot located in a room r_1 is supposed to approach a table t_1, located in room r_2 to fetch a bottle of water. A plan could comprise, moving to room r_2, approaching table t_1, and detecting a bottle of water on it. Lets also assume that another bottle of water gets into the field of view of the robot as it moves towards room r_2. Iff the *bottle of water detector* is activated before approaching table t_1, it could be detected and the plan could be optimized using such bottle of water instead.

[2] Exogenous events are those which are not the result of a deliberate action of the robot, *e.g.*, a "low on battery" event.

Method	**Success Rate**
Multimodal Passive Learning Agent	**0.75**
TH + GoogleNet	0.35
TH + GoogleNet2	0.35
TH + AlexNet	0.35
TH + VGG16	0.6
MCG + GoogleNet	0.5
MCG + AlexNet	0.15
MCG + ResNet	0.55
MCG + VGG16	0.55
CS + GoogleNet	0.45
CS + GoogleNet2	0.6
CS + AlexNet	0.2
CS + VGG16	0.45
R-CNN	0.4

TABLE I
NORMALIZED SUCCESS RATE OF THE OBJECT SEARCH TEST OF THE MULTIMODAL PASSIVE LEARNING AGENT COMPARED TO DIFFERENT COMBINATIONS OF SEGMENTATION ALGORITHMS AND CNNS.

of thirteen convolutional layers (also with maxpool in between), followed by three fully connected layers.

- Regions with Convolutional Neural Network (R-CNN) [24] performs localization and classification of the objects in the image. It generates category-independent region proposals, then a convolutional network extracts a fixed-length feature vector from each region and finally the third module, which is a set of class-specific linear SVMs, scores each feature vector. Since it performs localization by itself no previous segmentation step was added to this network.

The trainings used for these networks are generic trainings provided by their authors. They have been combined with the following three main segmentation algorithms:

- Top-Hat [25], a morphology transformation based algorithm commonly used for segmentation purposes.
- Multiscale Combinatorial Grouping [26], an algorithm for bottom-up hierarchical image segmentation.
- The Cognitive Subtraction algorithm explained in Section III-A as the second layer of the passive learning agent.

To test the system it was asked to locate 20 objects selected randomly among the five tables setup explained at the beginning of this section (tables A, B, C, D and E). If the first choice of the algorithm was the correct table it was counted as a success, otherwise it was considered a failure. As it can be seen in the results of Table I, the multimodal passive learning agent outperforms the rest of the state-of-the-art solutions. The low quality of the images obtained in the passive learning process generate very bad results in regular segmentations and CNN combinations, while the semantic relationship that the agent uses helps improve its successful results.

B. Perceptive Planning Architecture Tests

Three methods are compared on these experiments: random selection, a traveling salesman policy, and the proposed perceptive planning architecture. The robot was asked to find each object from the five different tables and the success rate and time spent by the robot was measured. The success rates obtained in the experiments where 0.36, 0.35 and 0.9 for the random, traveling salesman and perceptive planning architecture solutions, respectively. The average time that the robot used to approach the correct table was of 82.68, 61.0 and 26.13 seconds for the random, traveling salesman and perceptive planning architecture solutions.

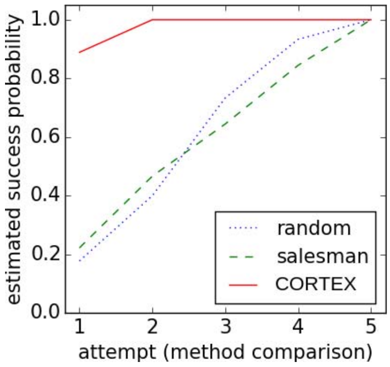

Fig. 4. Average success rate when trying to find the correct table for each of the three methods. I.e. the first attempt is the average of times the correct table was selected the first by the algorithm when trying to find the object.

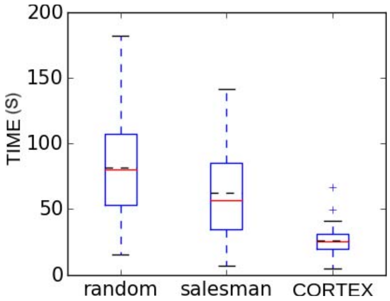

Fig. 5. Average time spent to find the object for each of the three methods tested.

Figure 4 shows a graphic specifying how the success rate evolved with the different methods as attempts are made for all the objects considered. It can be appreciated in Fig. 4 that the success rate evolves similarly as they keep trying for the random and traveling salesman policies while the performance of CORTEX using the passive learning agent was much better since the first attempt.

Figure 5 provides boxplots for the time spent by the robot using each of the methods to find the objects. Although random and traveling salesman have similar success rates they differ in time as the later policy always choose the shorter paths.

V. CONCLUSIONS AND FUTURE WORK

A new extension to the CORTEX architecture that provides an effective way to speed up object search has been presented. The improved functionality has been tested in a real scenario with a mobile manipulator.

The results shown in the experiments section demonstrate the effectiveness of the assumptions of the passive learning agent. The agent was able to successfully predict the location of objects, outperforming state-of-the-art CNN algorithms. On the other hand, the results on the whole architecture that integrates the scene labeling verification step in CORTEX where also positive. The architecture outperformed other two table approaching techniques, both in success rate and in time taken to find the object.

As future a work, it would be interesting to integrate the search task as part of a more complex plan. Testing the solution with a higher level process that involves more actions would help us provide the demonstration of the real utility of the perceptive planning architecture. Currently, the architecture only supports one search at a time, but when higher tasks are taken into account it might be interesting to make the robot able to search and reason about more than one object at a time. Also, the distance to the different tables is not taken into account, for future releases of the system this information could be used as an input to help the robot take a better decision on which place to visit next.

Misplaced objects make the robot fail on its initial attempts to fetch them. It would be interesting to develop an extra passive mechanism able to spot these misplaced object and take them as an exception when performing the object search.

Finally, the agent integrated in the architecture is specific for objects lying on tables. A good improvement would be to enhance this agent to automatically detect the object container and to proceed accordingly, *i.e.*, changing the pipeline of the CS layer accordingly.

ACKNOWLEDGMENT

This work has been partially supported by the MICINN Project TIN2015-65686-C5-5-R, by the Extremaduran Goverment project GR15120, by the Red de Excelencia "Red de Agentes Físicos" TIN2015-71693-REDT, and by FEDER project 0043-EUROAGE-4-E (Interreg V-A POCTEP Program). The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of GPU hardware used for this research.

REFERENCES

[1] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, and M. Tenorth, "Robotic roommates making pancakes," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 529–536.

[2] Y. Kita, F. Kanehiro, T. Ueshiba, and N. Kita, "Strategy for folding clothing on the basis of deformable models," in *Image Analysis and Recognition: 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part II*, A. Campilho and M. Kamel, Eds. Cham: Springer International Publishing, 2014, pp. 442–452.

[3] P. Bustos, L. Manso, J. Bandera, A. Romero-Garcés, L. Calderita, R. Marfil, and A. Bandera, "A unified internal representation of the outer world for social robotics," in *Advances in Intelligent Systems and Computing*, 2016, vol. 418.

[4] M. Lorbach, S. Höfer, and O. Brock, "Prior-assisted propagation of spatial information for object search," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 2904–2909.

[5] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3515–3522.

[6] L. Kunze, K. K. Doreswamy, and N. Hawes, "Using qualitative spatial relations for indirect object search," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 163–168.

[7] M. A. Gutiérrez, L. J. Manso, H. Pandya, and P. Núñez, "A passive learning sensor architecture for multimodal image labeling: An application for social robots," *Sensors*, vol. 17, no. 2, p. 353, 2017.

[8] C. Wu, I. Lenz, and A. Saxena, "Hierarchical semantic labeling for task-relevant rgb-d perception." in *Robotics: science and systems*, 2014.

[9] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[10] R. B. Rusu, Z. C. Marton, N. Blodow, A. Holzbach, and M. Beetz, "Model-based and learned semantic object labeling in 3d point cloud maps of kitchen environments," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 3601–3608.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[12] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sept. 1999. [Online]. Available: http://doi.acm.org/10.1145/331499.331504

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[15] T. Mikolov and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, 2013.

[16] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[17] L. Manso, P. Bustos, P. Bachiller, and P. Núñez, "A perception-aware architecture for autonomous robots," *International Journal of Advanced Robotic Systems*, vol. 12, no. 174, p. 13, 2015.

[18] L. Manso, L. Calderita, P. Bustos, and A. Bandera, "Use and advances in the active grammar-based modeling architecture," *Málaga, Spain-June 2016*, p. 25.

[19] L. J. Manso, R. Cintas, P. Bustos, and C. T. Deptartment, "Improving the lifecycle of robotics components using Domain-Specific Languages," in *Second International Workshop on Domain-Specific Languages and models for ROBotics systems, DSLRob'11*, 2011.

[20] R. Marfil, L. V. Calderita, J. P. Bandera, L. J. Manso, and A. Bandera, "Toward Social Cognition in Robotics : Extracting and Internalizing Meaning from Perception," in *Workshop of Physical Agents WAF2014*, no. June, Leon, Spain, 2014, pp. 1–12.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, Jan 2016.

[25] M. F, "Contrast feature extraction." *Cherman JL (ed) Analyse quantitative des microstructures en sciences des materiaux, biologie et medecine, Rieder, Stuttgart,*, p. 374, 1977.

[26] J. Pont-Tuset, P. Arbelaez, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2016.