Draft Version. Final version published in the 17th International Conference on Autonomous Robot Systems and Competitions (ICARSC)

Online Context-based Object Recognition for Mobile Robots

J.R. Ruiz-Sarmiento*, Martin Günther†, Cipriano Galindo*, Javier González-Jiménez* and Joachim Hertzberg†‡

*Dept. of System Eng. and Automation, Instituto de Investigación Biomédica de Málaga (IBIMA), University of Málaga, Spain
{jotaraul,cgalindo,javiergonzalez}@uma.es
†DFKI Robotics Innovation Center, Osnabrück Branch, 49076 Osnabrück, Germany
{martin.guenther, joachim.hertzberg}@dfki.de
‡Institute of Computer Science, Osnabrück University, Albrechtstr. 28, 49076 Osnabrück, Germany

***Abstract*—This work proposes a *robotic object recognition system* that takes advantage of the contextual information latent in human-like environments in an *online* fashion. To fully leverage context, it is needed perceptual information from (at least) a portion of the scene containing the objects of interest, which could not be entirely covered by just an one-shot sensor observation. Information from a larger portion of the scenario could still be considered by progressively registering observations, but this approach experiences difficulties under some circumstances, *e.g.* limited and heavily demanded computational resources, dynamic environments, etc. Instead of this, the proposed recognition system relies on an *anchoring process* for the fast registration and propagation of objects' features and locations beyond the current sensor frustum. In this way, the system builds a graph-based world model containing the objects in the scenario (both in the current and previously perceived shots), which is exploited by a *Probabilistic Graphical Model* (PGM) in order to leverage contextual information during recognition. We also propose a novel way to include the outcome of local object recognition methods in the PGM, which results in a decrease in the usually high CRF learning complexity. A demonstration of our proposal has been conducted employing a dataset captured by a mobile robot from restaurant-like settings, showing promising results.**

I. Introduction

Nowadays, object recognition systems tend to incorporate contextual information between objects, which has proven to increase the performance of *local* object recognition methods, *i.e.*, those that *only* rely on features of the objects themselves (such as their geometry or appearance), and neglect the intrinsic relations among objects in the scene [1]. Let's consider a classic scenario from the Artificial Intelligence and Robotics fields consisting of a waiter robot checking the tables' configuration in a restaurant. This contextual information can guide the recognition process stating that a long, thin object to the left of a plate is more probable to be a fork than a spoon, since that is the common, preferable configuration.

A large, growing body of literature has resorted to the Probabilistic Graphical Models (PGMs) framework [2] for modeling contextual relations [3–10]. In this framework, a set of *weights* are learned in a supervised training process [11], and then exploited by probabilistic inference to categorize and recognize sensory data. Applied to object recognition, training weights are associated with the different object classes (*e.g.*, mug, vase, milk-pot, etc.), and the features used to characterize them (*e.g.*, color, height, size, etc.) and their contextual relations (*e.g.*, distance between two objects, relative position with regard to a supporting surface, etc.).

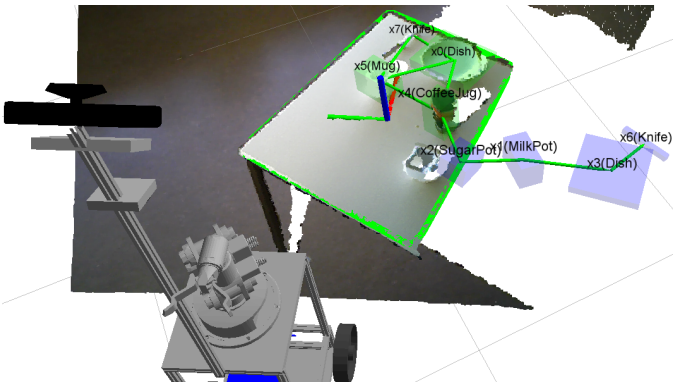


Fig. 1: Robot observing a partial view of a tabletop. Notice that only a part of the table is captured, but the world state model used by the object anchoring process still retains previously observed objects and relations.

Most works address the problem through *one-shot* recognition systems [3], [4], [9], [10], which recognize objects relying on single observations of the scene (in the form of RGB, depth or RGB-D images). Regarding the exploitation of contextual information, one-shot systems are seriously limited by the sensor frustum and possible occlusions, given that they are able to observe only a portion of the objects and relations appearing in the inspected scene. Some approaches cope with this issue by registering a number of observations prior to the recognition process in order to obtain a wider view of the scene [5–10]. However, the time and computational resources needed for gathering and registering such observations prevents their use in most robotic applications. Less attention has been paid to *online recognition* methods, which can mitigate these drawbacks by incorporating and exploiting objects and contextual relations not appearing in the current sensor observations, but previously perceived by the robot.

The contribution of this work is twofold. First, we present an *online object recognition system* that handles contextual information beyond the sensor frustum in an *online* fashion. For achieving that, we rely on an *anchoring process* [12] for fast registering and propagating the location and features of the perceived objects over time, even when they are out of the current camera field of view (see Figure 1). The output of the anchoring process is used to build and update a graph-based representation of the world, which combined with a *Conditional Random Field* (CRF) [2], a particular type of PGM, permit the system to fully exploit the contextual information in the environment by also considering that from past observed areas.

Our second contribution tackles the problem of the PGM training complexity, which seriously increases when the number of weights employed to comprehensively model the problem is large [11]. For that, we modify the usual CRF formulation to include the results of any local, off-the-shelf object recognition method able to provide a *confidence* measure of its results. Local methods can specialize in dealing with some kind of *low level* features, relieving the burden of learning their related weights, hence decreasing the training complexity. It is worth mentioning that the difference with previous works in the literature stems from the possibility to also model complementary, possibly higher-level features that are not used by local methods.

The next section puts our work in context, while Section III introduces CRFs and their application to object recognition. In Section IV, we describe the proposed *online* object recognition system, and experimental results are presented in Section V. Finally, Section VI outlines some conclusions and future work.

II. Related Work

Several object recognition systems can be found in the literature relying on geometric or appearance features of objects, like SIFT based approaches [13], bag of features models [14] or methods based on CAD model matching [15]. However, these methods yield ambiguous results under some circumstances, a drawback that can be alleviated with the use of contextual information [1].

The Probabilistic Graphical Models (PGMs) framework [2] is widely used for modeling and exploiting this kind of information. For example, the work in [5] proposes a model isomorphic to a Markov Random Field (MRF) and a rich set of features to represent the scene objects and their relations, while in [3], MRFs are combined with segmentation trees for the recognition of objects. There are also examples relying on Conditional Random Fields (CRFs), like [6], to classify objects into four categories: wall, floor, ceiling and clutter, and the work presented in [7], where the faces of a triangular mesh representing the scene are assigned to object classes. CRFs are also used in [8], [9], and [10], in conjunction with common-sense information codified into an ontology, for the recognition of objects appearing in office and domestic scenes, and in [4] for the modeling of context in RGB images. However, despite of the effort that has been made for properly modeling and exploiting objects' contextual information, less attention has been paid to their applicability to different mobile robot tasks [16], which probably imposes computational and/or time execution constrains.

Some other works, in addition to the use of contextual information, also expand the recognition results of observations over time. For example, in [17] a CRF is used for modeling both objects appearing in a scene and human activities over a set of observations. In [18] a pixel level semantic segmentation is performed through probabilistic inference over a CRF enforcing temporal consistency of segments between consecutive observations, resulting in a highly demanding computational task. A similar approach is presented in [19] with a temporal window of one observation. In this work we rely on an anchoring process to expand the locations and features of previously recognized objects over time, without the restriction of a temporal window. Anchoring [12] has been previously used to maintain a coherent representation of the robot surroundings [20], [21], which in addition is useful to perform efficient high-level robotic planning.

III. Applying CRFs to Scene Object Recognition

Probabilistic Graphical Models (PGMs) [2] have been used to compactly and efficiently exploit contextual relations between random variables. Applied to scene object recognition, where the aim is to assign to each of the scene objects their respective class (*e.g.*, mug, dish, spoon), such a problem is modeled as follows. Let $\boldsymbol{x} = \{x_1, .., x_n\}$ be a set representing n observed objects within a given scene, where each object x_i is characterized through a vector of m features $\boldsymbol{f}_{x_i u} = [f_{x_i u_1}, .., f_{x_i u_m}]^T$, *e.g.*, size, height or elongation, $L = \{l_1, .., l_k\}$ the set of the k possible object classes, and $\boldsymbol{y} = \{y_i, .., y_n\}$ a set of discrete random variables over L, that assign to each object in $\boldsymbol{x}$ a class from L. Thus, the scene object recognition problem, modeled by a CRF, consist of maximizing the probability distribution $P(\boldsymbol{y}|\boldsymbol{x})$, *i.e.*, to find the most probable classes assignment to $\boldsymbol{y}$ given the characterized objects' observations in $\boldsymbol{x}$.

A CRF is represented through a graph structure $H = (V, E)$, where V is a set of nodes representing random variables, and E a set of edges linking related nodes. Concretely, in the scene object recognition problem, each variable in $\boldsymbol{y}$ introduces a node in V, and two contextually related variables, *i.e.*, variables whose associated objects are close to each other in the scene, set an edge in E between their respective nodes. Then, according to the Hammersley-Clifford theorem [2], a number of functions called factors are defined over parts of H, encoding each one a piece of $P(\boldsymbol{y}|\boldsymbol{x})$. In this work we rely on two factor types: *unary*, related to nodes, and *pairwise*, associated with edges. The insight behind this is that unary factors encode the likelihood of a variable y_i to be assigned to a certain class l_i given the characterized object x_i, while pairwise factors express the compatibility of two related variables belonging to a certain pair of classes.

Concretely, unary factors $U(\cdot)$, and pairwise factors $I(\cdot)$, are defined by linear classification models as follows:

$$U(y_i, x_i, \boldsymbol{\omega}) = \sum_{l \in L} \delta(y_i = l) \boldsymbol{\omega}_l f(x_i) \quad (1)$$

$$I(y_i, y_j, x_i, x_j, \boldsymbol{\theta}) = \sum_{l_1 \in L} \sum_{l_2 \in L} \delta(y_i = l_1)\delta(y_j = l_2) \boldsymbol{\theta}_{l_1,l_2} g(x_i, x_j) \quad (2)$$

where $f(x_i)$ is the function that computes the vector of features $\boldsymbol{f}_{x_i u}$ of the object x_i, $g(x_i, x_j)$ provides the pairwise features $\boldsymbol{f}_{x_i x_j p} = [f_{x_i x_j p_1}, .., f_{x_i x_j p_q}]^T$ for objects x_i and x_j (*e.g.* perpendicularity, size ratio, etc.), $\boldsymbol{\omega}_l = [\omega_{1,l}, .., \omega_{m,l}]$ and $\boldsymbol{\theta}_{l_1,l_2} = [\theta_{1,l_1,l_2}, .., \theta_{q,l_1,l_2}]$ are vectors of weights associated to the class l and the combination of classes l_1 and l_2 respectively, both learned during the CRF training, and $\delta(y_i = l)$ is the Kronecker delta function.

Once these factors have been defined, the computation of $P(\boldsymbol{y}|\boldsymbol{x})$ can be expressed by means of log-linear models as:

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\theta})} e^{-\epsilon(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\theta})} \quad (3)$$

where $Z(\cdot)$ is the partition function, which plays a normalizing role so that $\sum_{\xi(\boldsymbol{y})} P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\theta}) = 1$, with $\xi(\boldsymbol{y})$ being a possible assignment to the variables in $\boldsymbol{y}$, and $\epsilon(\cdot)$ is the so-called energy function defined as the sum of all the factors defined over the graph:

$$\epsilon(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\theta}) = \sum_{i \in V} U(y_i, x_i, \boldsymbol{\omega}) + \sum_{(i,j) \in E} I(y_i, y_j, x_i, x_j, \boldsymbol{\theta}) \quad (4)$$

The CRF training yields the weights $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ that maximize the likelihood function:

$$\max_{\boldsymbol{\omega}, \boldsymbol{\theta}} L_P(\boldsymbol{\omega}, \boldsymbol{\theta} : D) = \max_{\boldsymbol{\omega}, \boldsymbol{\theta}} \prod_{d \in D} P(y_d|x_d) \quad (5)$$

where D is the set of all the scenes used for training, x_d is a set containing the characterized objects in the scene $d \in D$, and y_d are their corresponding ground truth classes [11]. Since solving Eq. 5 requires the computation of the partition function, which is intractable in practice, usually an approximated function replaces it. We have chosen the pseudo-likelihood one in this work [2].

Despite this simplification, the learning process remains complex if the number of considered object classes $|L|$ and features used to describe them $\left|\boldsymbol{f}_{x_i u}\right|$ and their relations $\left|\boldsymbol{f}_{x_i x_j p}\right|$ is high, which results in a large number of weights as well. On the other hand, it is desirable to account for a comprehensive variety of features in order to properly characterize both objects and relations. In Sec. IV-C2 we describe our approach to tackle this issue.

Once these weights are learnt, and provided a graph representation $H = (V, E)$ of the objects and relations appearing within a given scene (see Sec. IV-C1), a probabilistic inference process over the resultant CRF predicts the objects' most probable classes, as described in Sec. IV-C3.

IV. Context-Aware Online Object Recognition

The proposed recognition system is a combination of: i) a local object recognition method, ii) an anchoring process, and iii) a Conditional Random Field (see Figure 2). In a nutshell, the *local object recognition method*[1] (Sec. IV-A) segments the sensor point cloud into object clusters and yields a recognition result for each one in the form of vector of confidence values. This is the input for the *anchoring process* (Sec. IV-B), which tries to anchor the newly detected objects to previously observed ones, and supports the creation of a scene graph comprising all the currently observed objects and nearby ones that are out of the camera frustum, with edges connecting objects that are close to each other. A CRF is then built incorporating the nodes and edges of the scene graph along with their features, as well as the confidence values from the local object recognition (Sec. IV-C). Finally, a probabilistic inference process over the CRF yields the scene object recognition results.

A. Segmentation, Tracking and Local Object Recognition

The first steps in our processing pipeline perform the segmentation of the current sensor (RGB-D) data and a local object recognition, which classifies each segmented object separately. Here, we use the state-of-the-art spin-image based object recognition method by Oliveira *et al.* [22]. Its output is a set of positions and bounding boxes for each object, along with a vector $\boldsymbol{c}_{x_i} = [c_{l_1}, .., c_{l_k}]$ of confidence values, representing the confidence with which object x_i belongs to the respective class in L. This output is used both by the anchoring process (Sec. IV-B) and the CRF (Sec. IV-C2).

Apart from segmentation and local object recognition, Oliveira et al.'s method [22] also provides real-time tracking of the recognized objects: a unique *track ID* is attached to all observations of the object in subsequent camera frames. However, this tracking is lost when the object is no longer in view of the RGB-D camera.

Notice that the proposed system can integrate any off-the-shelf local object recognition method yielding a bounding box and a confidence vector for each object. Although we make use of the tracking functionality in Sec. IV-B, our system does not require it. In fact, we have also used the system to process the output from the ROS tabletop_object_detector [23] recognition method, which does not provide track IDs[2].

B. Object Anchoring and World Modeling

The anchoring process keeps a persistent world model of the locations, features, identities and classes of the objects perceived so far. This enables the CRF to exploit context between currently observed objects and previous ones which are outside the current camera view (Figure 3 shows an example situation).

[1]In this paper, we use the term *local* object recognition for this process to distinguish it from the presented, complete object recognition system, which does *joint* classification of all objects in the scene.

[2]We have not used that detector in the context of this paper since it requires CAD models of all objects and can only handle rotationally symmetric objects.

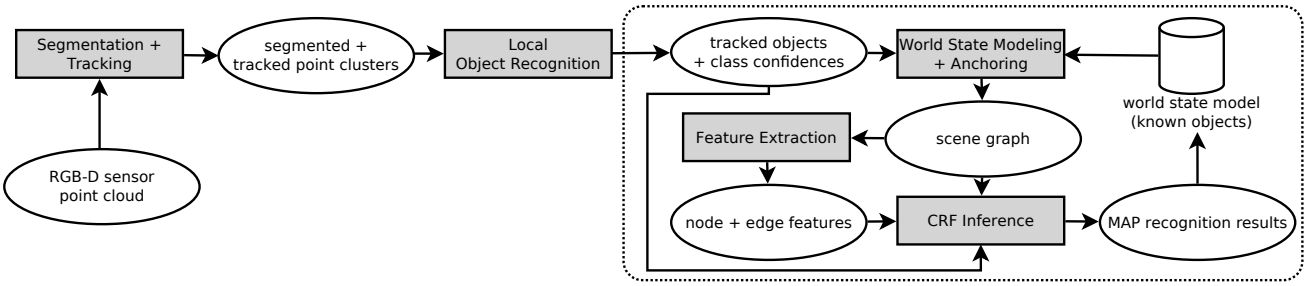


Fig. 2: Overview of the proposed system. Ovals represent consumed/produced data, while boxes stand for processes. The contribution of this paper is highlighted by the dotted box. Locally tracked objects are anchored to previously observed ones, and a scene graph of objects is built including both currently tracked objects and nearby known ones out of view of the camera. All objects in the graph are jointly classified by an inference process over a CRF, and the set of known objects is updated.

Whenever a new set of objects is reported by the local object recognition, the following steps are executed:
1) Anchoring attempts to match the new objects to all nearby known ones from the current world model. In order to do so, the similarity between pairs of objects is calculated by a Support Vector Machine (SVM) that was trained on the distance, size and difference of the two objects' class confidence vectors c_i, c_j. Next, the best assignment of all objects is calculated by the Hungarian method [24], using the computed similarity as a cost function. If the cost of assigning an object exceeds a certain threshold, a new object is inserted into the world model; otherwise, the object representation is updated. If the local recognition system supports tracking and the object is already tracked, its track ID will be used to instantly match the new observation to its representation in the world model.
2) A scene graph for the local scene is created by first adding one node for each currently observed object, then recursively adding nodes for objects that are closer than a certain context range to a node already in the graph (which can include known objects that are close to currently observed objects, but outside the camera frustum). An edge is added to the graph between each pair of nodes that are within context range of each other (see Figure 3).
3) After computing the objects' MAP classes (see following section), the world state model is updated with the result.

C. CRF Building and Inference

Given the scene graph in the world model, a CRF model is built according to it and the features and relations of its constituent objects (Sec. IV-C1), which also integrates the confidence vector from the local object recognition method (Sec. IV-C2). Finally, an inference process over the CRF computes the most probable classes of all objects (Sec. IV-C3).

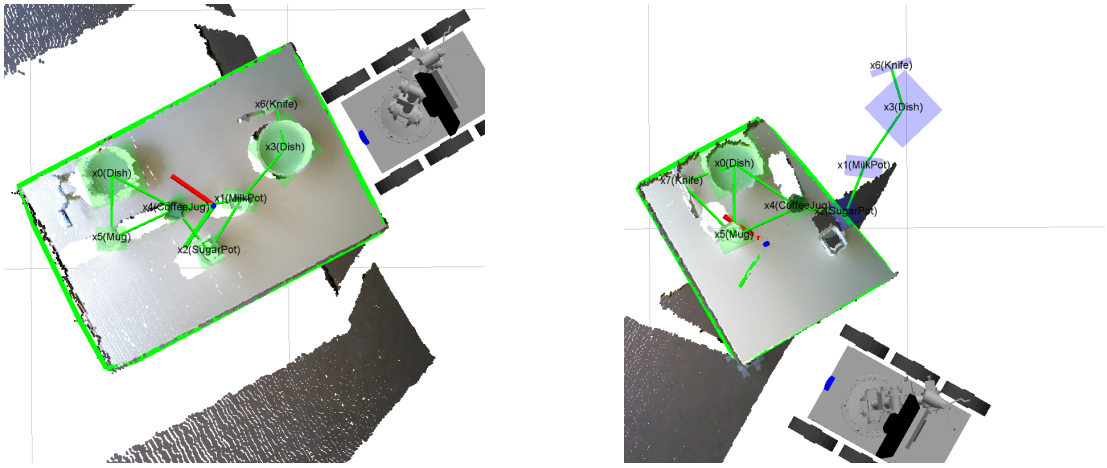
1) Graph building and Feature Extraction: The building of the CRF graph structure $H = (V, E)$ is straightforward, since the anchoring process already provides the set of nodes V, each one associated with a random variable from $\boldsymbol{y}$ representing the class of a scene object x_i, and the set of edges/relations E between those objects. Note that a scene object x_i can be an object present in the current sensor observation, or a closer one out of the sensor view but previously detected.

These objects and relations are subsequently characterized through the vectors of features $\boldsymbol{f}_{x_i u}$ and $\boldsymbol{f}_{x_i x_j p}$ respectively, which are integrated into the CRF model as introduced in Eq. 1 and Eq. 2. The (unary) features used in this work to describe an object are: volume, horizontal and vertical area, horizontal and vertical elongation, and distances from the estimated table center and table border. The (pairwise) features describing objects' relations are: ratio between the objects' volumes, difference in height above ground, horizontal distance between their centroids, difference in distances from table center/border, and angle between the objects and the table center. Also a bias term is included to allow the CRF to consider the co-occurrence probability between object classes.

2) Integrating local object recognition results: Our CRF model can be enriched with the outcome from any local object recognition method (or a combination of them) able to provide a confidence vector of its results, which permits the recognition system to take advantage of methods exploiting specialized feature descriptors. For that, we introduce the confidence vector c_{x_i} of an object x_i into the usual CRF unary factor formulation (see Eq. 1) in the following way:

$$U(y_i, x_i, \boldsymbol{\omega}) = \sum_{l \in L} \delta(y_i = l) \boldsymbol{\omega}_l f(x_i) c_{x_i}(l) \tag{6}$$

Thus, the components of the confidence vector play the role of an additional feature, not related to any weight, that have a different value for each object class. This integration also leads to a reduction in the number of weights present during the CRF learning, alleviating the training complexity. In this way, the CRF focuses on exploiting high-level features of objects and relations, and releases the work with specialized feature descriptors to the local method, which modeling into the CRF typically involves a large number of weights. For example, in our experiments where we have considered 9 different object classes, the number of weights associated with features of objects is $\left|\boldsymbol{f}_{x_i u}\right| \cdot |L| = 7 \cdot 9 = 63$, while the number of those related to contextual features results $\left|\boldsymbol{f}_{x_i, x_j p}\right| \cdot |L|^2 = 6 \cdot 81 = 486$, giving a total of 549 weights. The employed local recognition method represents each object by a set of 10 shape features, each one with a descriptor vector of 45 components, resulting in a total descriptor length



(a) robot observing a tabletop scene

(b) a later observation of the same scene from a different viewpoint

Fig. 3: Top-down views of a robot observing a tabletop scene. Transparent boxes: bounding boxes of anchored objects with their CRF classification result (CRF nodes; green: currently tracked, blue: not currently tracked); lines: context edges between nearby objects (CRF edges). The bounding polygon of the currently observed table surface and its center are also shown. Note: Figure 3b is an orthographic projection of the situation depicted in Figure 1.

of 450. Thus, their modeling into a CRF would suppose the addition of $450 \cdot |L| = 4050$ weights, which clearly would increase the training complexity, even dropping the recognition performance.

3) Probabilistic inference: Once the CRF has been modeled including: a graph representation $H = (V, E)$, the extracted features of the graph components, and the confidence values from the local object recognition method, a probabilistic inference process over such a CRF is in charge of providing the recognition outcome. Concretely, the objects' recognition consists of finding the classes assignation $\hat{\boldsymbol{y}}$ that maximizes the probability distribution $P(\boldsymbol{y}|\boldsymbol{x})$, that is:

$$\hat{\boldsymbol{y}} = \arg\max_{y} P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{\theta}) \tag{7}$$

This calculus is commonly referred as the Maximum a Posteriori problem (MAP), which in this work is carried out by means of the Iterated Conditional Modes method [25], an approximated solution that mitigates the heavy computational burden required by exact approaches. We have resorted to the implementation of the aforementioned training and inference processes within the UPGMpp library [26].

V. Experimental Results

To test our system, we collected a data set of 15 scenes of a robot equipped with a RGB-D camera driving around a table and turning towards it from different locations. The table contained a number of objects in varying table settings. In total, the data set contains 1387 seconds of observation and 144 unique objects from 9 object classes [3]. We have followed a five-fold cross-validation process, training the CRF on twelve of the scenes while using the other three for testing, and then switching the folds. This data set was intentionally chosen because of two challenging characteristics: on the one hand, the small objects (fork, knife and spoon) only produce a small set of points in the captured point clouds; they are almost the same size and contain reflective parts, so only the handle is actually visible. This combination makes them hard to be distinguished based only on local features. On the other hand, the robot normally sees only a part of the scene, which means that the full object context information is not available from the current sensor data (see Figure 3).

[3]Available at https://goo.gl/BQiXC8

TABLE I: Classification accuracy on sub-data sets ("small objects": Fork, Knife, Spoon). *Loc*: local object recognition only (base line [22]); *CRF+A*: CRF with objects outside the field of view supplied by Anchoring; *CRF+Loc*: CRF integrated with local object recognition; *CRF+A+Loc*: complete, proposed system (CRF, anchoring, local object recognition).

	Loc	CRF+A	CRF+Loc	CRF+A+Loc
small objects	46.32 %	75.00 %	72.58 %	75.54 %
other objects	91.15 %	90.73 %	94.38 %	93.15 %
total	76.64 %	85.05 %	85.43 %	86.82 %

The results are summarized in Table I. As expected, recognizing small objects yields poor accuracy results when using only the local object recognition method. However, our combined system exploits context to achieve a significant improvement (29.22 % increase in accuracy). The results also demonstrate the performance boost by using context with objects outside the current field of view (CRF+A, Sec. IV-B) and integrating local object recognition results into the CRF (CRF+Loc, Section IV-C). In total, the complete combined system (CRF+A+Loc) achieved an increase of 10.18 % in accuracy over the employed local object recognition method [22]. Figure 4 shows the aggregated confusion

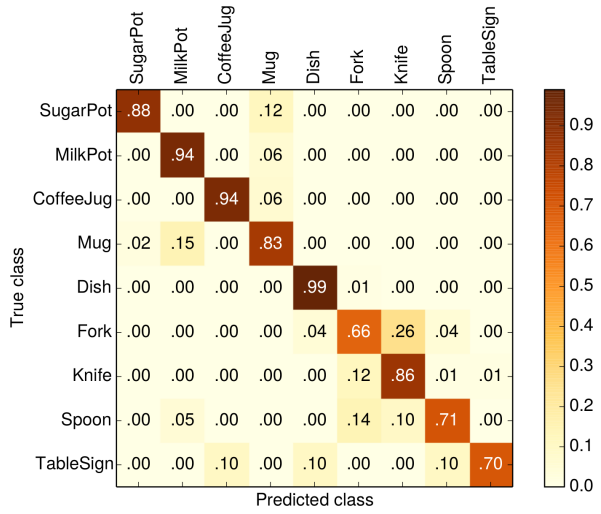


Fig. 4: Confusion matrix of the combined method (CRF+A+Loc in Table I) – classification accuracy: 86.82 %.

matrix yielded by the proposed system over the data set.

Regarding computational costs, training, which has to be completed only once, took on average 1421.5 s per fold. The runtime of feature extraction was 0.064 s, and for MAP inference 0.005 s per scene graph (excluding time for the local object recognition). All experiments have been performed on a standard laptop (2.6 GHz Core i7 CPU, 8 GB RAM).

VI. Conclusions

This paper has presented an *online object recognition system* that exploits contextual relations between the scene objects beyond the sensor field of view. This is achieved through the use of an *anchoring process*, which in an *online* fashion propagates the features and locations of previously perceived objects over time, and a *Conditional Random Field* (CRF) that captures the scene objects' relationships and enables their exploitation by means of a probabilistic inference process. We have also tackled the problem of the CRF training complexity by proposing a new formulation that leverages the confidence results provided by an off-the-shelf object recognition method, which specializes in dealing with low-level features. The conducted evaluation supports our claims: i) the use of contextual information has improved the recognition results yielded by a state-of-the-art local recognition method, ii) the integration of the outcome of such method into the proposed CRF formulation has shown a positive effect on performance, and iii) the inclusion of information previously perceived by the robot through anchoring leads to further performance improvements.

Acknowledgment

This work is supported by the European projects RACE [Call:FP7-ICT-2011-7, contract number: 287752], MoveCare [Call:H2020-ICT-2016-1, contract number: 732158], the Spanish grant program FPU-MICINN 2010 and the PROMOVE project [ref:DPI2014-55826-R].

References

[1] C. Galleguillos and S. Belongie, "Context based object categorization: A critical survey," *Comput. Vis. Image Underst.*, vol. 114, no. 6, pp. 712–722, Jun. 2010.

[2] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[3] X. Ren, L. Bo, and D. Fox, "RGB-(D) scene labeling: Features and algorithms," in *CVPR*, Jun. 2012, pp. 2759–2766.

[4] Y. Xiang, X. Zhou, Z. Liu, T.-S. Chua, and C.-W. Ngo, "Semantic context modeling with maximal margin conditional random fields for automatic image annotation," in *CVPR, 2010*, 2010, pp. 3368–3375.

[5] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena, "Contextually guided semantic labeling and search for three-dimensional point clouds," *Int. J. Rob. Res.*, vol. 32, no. 1, pp. 19–34, 2013.

[6] X. Xiong and D. Huber, "Using context to create semantic 3D models of indoor environments," in *BMVC*, Sep. 2010.

[7] J. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. Torr, "Mesh based semantic modelling for indoor and outdoor scenes," in *CVPR*, Jun. 2013, pp. 2067–2074.

[8] J. R. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez, "Building multiversal semantic maps for mobile robot operation," *Knowl.-Based Syst.*, vol. 119, pp. 257–272, 2017.

[9] J. R. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez, "Exploiting semantic knowledge for robot object recognition," *Knowl.-Based Syst.*, vol. 86, pp. 131–142, 2015.

[10] J. R. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez, "Scene object recognition for mobile robots through semantic knowledge and probabilistic graphical models," *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8805–8816, 2015.

[11] J. R. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez, "A survey on learning approaches for undirected graphical models. application to scene object recognition," *Int. J. Approx. Reason.*, vol. 83, pp. 434–451, Apr. 2016.

[12] S. Coradeschi and A. Saffiotti, "An introduction to the anchoring problem," *Robot. Auton. Syst.*, vol. 43, no. 2-3, pp. 85–96, 2003.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, vol. 2, 2006, pp. 2161–2168.

[15] M. Günther, T. Wiemann, S. Albrecht, and J. Hertzberg, "Model-based furniture recognition for building semantic object maps," *Artif. Intell.*, 2015, DOI 10.1016/j.artint.2014.12.007.

[16] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robot. Auton. Syst.*, vol. 66, pp. 86–103, 2015.

[17] H. S. Koppula and A. Saxena, "Learning spatio-temporal structure from RGB-D videos for human activity detection and anticipation," in *ICML*, 2013.

[18] G. Floros and B. Leibe, "Joint 2D-3D temporally consistent semantic segmentation of street scenes," in *CVPR*, Jun. 2012, pp. 2823–2830.

[19] R. de Nijs, S. Ramos, G. Roig, X. Boix, L. Gool, and K. Kühnlenz, "On-line semantic perception using uncertainty," in *IROS*, Oct. 2012, pp. 4185–4191.

[20] N. Blodow, D. Jain, Z.-C. Marton, and M. Beetz, "Perception and probabilistic anchoring for dynamic world state logging," in *Humanoids*, 2010, pp. 160–166.

[21] J. Elfring, S. van den Dries, M. J. G. van de Molengraft, and M. Steinbuch, "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robot. Auton. Syst.*, vol. 61, no. 2, pp. 95–105, 2013.

[22] M. Oliveira, G. H. Lim, L. Seabra Lopes, H. Kasaei, A. Tome, and A. Chauhan, "A perceptual memory system for grounding semantic representations in intelligent service robots," in *IROS*, 2014.

[23] M. T. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Sucan, "Towards reliable grasping and manipulation in household environments," in *ISER*, 2010, pp. 241–252.

[24] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist. Q.*, vol. 2, no. 1-2, pp. 83–97, 1955.

[25] J. Besag, "On the statistical analysis of dirty pictures," *Royal Statistical Society*, vol. Series B, no. 2, pp. 259–302, 1986.

[26] J. R. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez, "UPGMpp: a software library for contextual object recognition," in *Recognition and Action for Scene Understanding (REACTS)*, 2015.