PR2 Looking at Things – Ensemble Learning for Unstructured Information Processing with Markov Logic Networks

Daniel Nyga[1], Ferenc Balint-Benczedi[2] and Michael Beetz[2]

***Abstract*— We investigate the perception and reasoning task of answering queries about realistic scenes with objects of daily use perceived by a robot. A key problem implied by the task is the variety of perceivable properties of objects, such as their shape, texture, color, size, text pieces and logos, that go beyond the capabilities of individual state-of-the-art perception methods. A promising alternative is to employ combinations of more specialized perception methods. In this paper we propose a novel combination method, which structures perception in a two-step process, and apply this method in our object perception system. In a first step, specialized methods annotate detected object hypotheses with symbolic information pieces. In the second step, the given query Q is answered by inferring the conditional probability $P(Q \mid E)$, where E are the symbolic information pieces considered as evidence for the conditional probability. In this setting Q and E are part of a probabilistic model of scenes, objects and their annotations, which the perception method has beforehand learned a joint probability distribution of. Our proposed method has substantial advantages over alternative methods in terms of the generality of queries that can be answered, the generation of information that can actively guide perception, the ease of extension, the possibility of including additional kinds of evidences, and its potential for the realization of self-improving and -specializing perception systems. We show for object categorization, which is a subclass of the probabilistic inferences, that impressive categorization performance can be achieved combining the employed expert perception methods in a synergistic manner.**

I. Introduction

As autonomous robots are to perform manipulation tasks that are more and more complex in environments that are less and less structured we need to substantially scale their object perception skills. Ideally, robots have to be capable of perceiving any task relevant object in any task relevant context. One of the big challenges here is that in most situations the scenes that a robot has to perceptually interpret include objects with different perceptual characteristics. A scene on a breakfast table, for example, typically includes textured objects such as jelly jars and cereal boxes, objects characterized by their shapes such as bowls and cups, translucent objects such as glasses, and small objects such as knives and forks.

In the past, it has proven to be difficult to equip robots with perception algorithms that can handle objects with very different perceptual characteristics. In most cases, scenes were simplified to account for the perceptual capabilities of the robot.

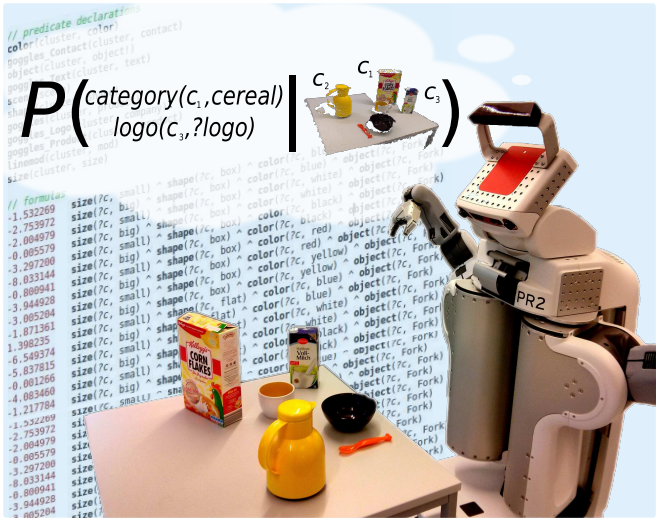

Fig. 1: PR2 looking at a breakfast table.

A promising alternative is the development of perception systems that are capable of synergetically combining more specialized algorithms to better scale towards natural environments and scenes.

In this paper, we propose a specific framework for object perception in scenes that include objects with different perceptual characteristics. Our approach implements object perception as a two step process. In the first step, specialized algorithms operate on perceived object hypotheses, extract perceptual information from the sensor data, and semantically annotate the respective object hypotheses with these pieces of information. In a second step, the symbolic annotations of objects and the whole scene are used as evidence to probabilistically infer the information that the robot requests from its perception system. To this end, the robot has previously learned a joint probability distribution over objects, their annotations, the co-occurrences of objects, and the occurrence of objects in different kinds of scenes.

The key idea of the proposed method is depicted in Figure 1. The robot asks queries, such as "is the category of the object hypothesis (the RGB-D point cluster) c_1 a cereal and what is the expected logo on the object hypothesis/point cluster c_3." These queries are transformed into relational con-

[1]D. Nyga is with the Intelligent Autonomous Systems Group, Department for Computer Science, Technische Universität München, Germany. nyga@cs.tum.edu

[2]F. Balint-Benczedi and M. Beetz are with the Institute for Artificial Intelligence and the TZI (Center for Computing Technologies), Universität Bremen, Germany. {balintbe, beetz}@cs.uni-bremen.de

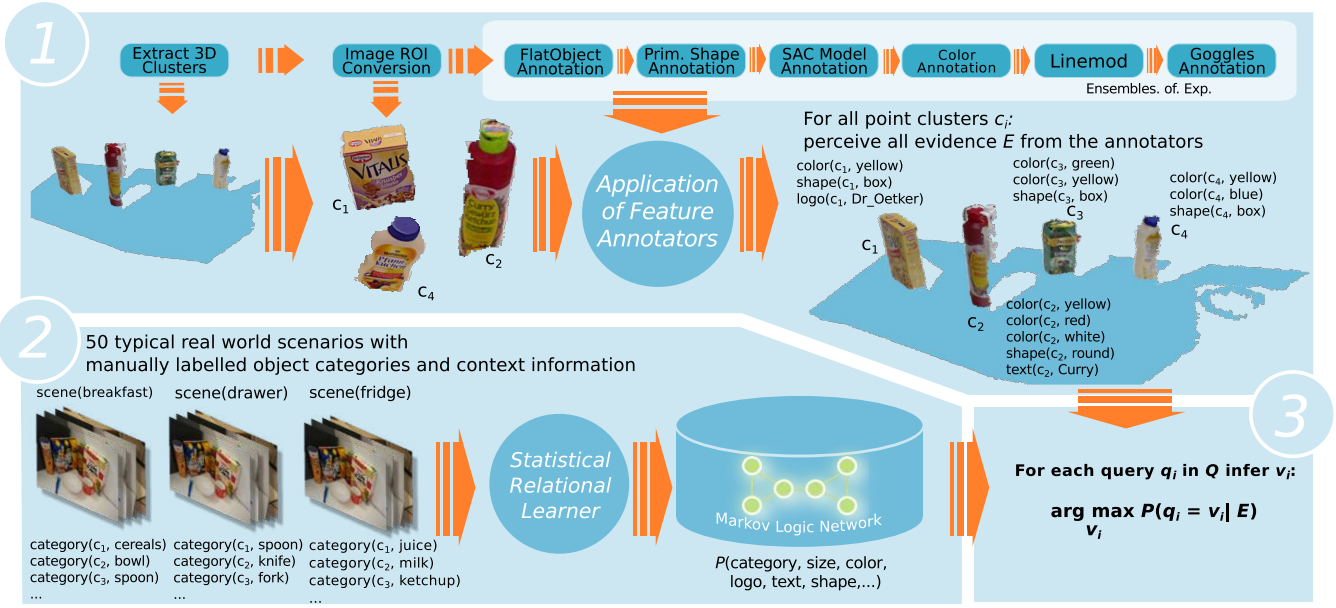

Fig. 2: Architecture of the system: (1) segmentation of point clouds into regions of interests, (2) the statistical relational learning and (3) reasoning system.

ditional probabilities *P(category(c_1,cereal),logo(c_3,?logo) | E)* that take the observed scene as their evidence.

In this paper, we propose an object perception system that extends the ROBOSHERLOCK[1] framework for unstructured information processing in robot perception. We believe that the formulation of object perception as a relational probabilistic reasoning problem has several advantages over alternative approaches. First, using perception algorithms for collecting perceptual evidence rather than making decisions makes the use of multiple specialized algorithms straightforward: they simply add their findings as annotations. Second, as inferences are drawn probabilistically on the basis of collected evidence, the system can also handle inconsistent annotations in a meaningful way. Third, the system can answer queries concerning all aspects contained in the probabilistic model under the given evidence. Fourth, the perception system can also exploit the regularities of the domain with respect to objects and their appearance and the occurrence/co-occurrence of objects in scenes. We validate our approach showing how combining very elementary perception mechanism can significantly boost recognition rates, while also demonstrating how our approach can be thought of as more than just a recognition framework.

The remainder of the paper is organized as follows. In Section II we describe the overall system and the annotators, followed by the presentation of how annotations are combined in Section III. Section IV offers a thorough evaluation of our proposed method. Finally, we look at some of the more recent advances in Section V, concluding in Section VI with future work and discussions.

[1]http://www.pr2-looking-at-things.com

II. OVERVIEW

We present a perception system that perceives scenes in prespecified regions of interest such as the top of counters, the content of drawers, and the content of fridges. It detects point clusters in these regions that might correspond to objects and treats these as being unstructured information, annotating each cluster with abstract information pieces such as the color, the size, the shape, the text, and the logo on the respective objects and uses these annotations as evidence E for probabilistic reasoning. Using a learned joint probability distribution over annotated scenes, we can then infer answers to perception-related questions Q by computing the conditional probability $P(Q \mid E)$.

Figure 2 gives a more detailed picture of the operation of our approach. It consists of three main components: (1) an image annotation component that segments point clouds into regions of interests that correspond to object hypotheses and annotates the individual hypotheses, (2) a statistical relational learning system that learns joint probability distributions over annotated scenes, and (3) the probabilistic reasoning system.

The image annotation component employs object segmentation mechanisms that detect objects on supporting planes, in drawers, and refrigerators. As the focus of this paper is the boosting of object perception through method combination, we restrict ourselves to scenes where the individual objects are clearly separable. Approaches that deal with object recognition in more cluttered scenes include [4], [5], [6]. Dealing with more cluttered scenes is on our future research agenda.

We employ simple Euclidean clustering in 3D space for objects located on a supporting plane and segmentation methods informed by semantic 3D object maps [3] that filter out everything except for the region of interest and generate

Annotator	Condition	MLN Predicate	Description
Color	Always	color(cluster, color)	The color annotator returns semantic color annotation based on color distribution in HSV color space. Colors: *blue, red, black, green, yellow, white*. Depending on the distribution, one object can have multiple colors.
Size	Always	size(cluster, size)	The size annotator classifies objects into *small* or *big* depending on distance between extreme points normalized with the distance to the camera. Values returned: *big, small*.
Goggles	If Google goggles returns text or logos	logo(cluster, logo) text(cluster, text) texture(cluster, t)	The annotator sends the image region of interest to the Google Goggles servers and parses the answer to extract text, logo, and texture information.
FlatObject	If there are objects that are too flat to be found by the general 3D clustering	shape(cluster, shape)	After extraction 3D clusters from the table this annotator looks for additional object hypotheses in color space (e.g., napkins, ...).
PrimShape	Always	shape(cluster, shape)	This annotator fits lines and circles to 3D point clusters projected on to the 2D plane using RANSAC [1]. Values returned: *box, round*
LineMod	Confidence that c is one of the objects looked for exceeds threshold	linemod(cluster, category)	This annotator matches each object hypothesis to a set of object models that the robot should actively look for using the Linemod algorithm [2].
SACmodel	If enough inliers for a model are found	shape(cluster, shape)	This annotator recognizes cylindrical objects in 3D space. If the number of inliers found exceeds the given threshold (60% of the total points in a cluster) the annotator accepts the match. Value returned: *cylinder*
Location	Always	scene(scene!)	This annotator interprets object positions in terms of a semantic environment map [3] and returns places such as *counter tops, tables, fridges, and drawers.* The *depth*- and *RGB*-image are being filtered leaving only pixels in a pre-defined region of interest.

TABLE I: Description of the annotators, the conditions under which they work and the predicate declarations in the MLN.

object hypotheses in the remaining region of interest. As some annotators (SacModel, Size, PrimitiveShape, etc.) use point clouds ([7]) as their input whilst others use RGB images (Linemod, Color, Goggles), a converter is used to find the region of interest corresponding to the extracted clusters. Having a representation of object candidates both in 3D as well as in image space all annotators can be run on the object hypotheses in order to attach semantic information to those.

Annotators are specialized perception routines that perceive specific aspects of information. For example, the color annotator asserts the fact *color(c,col)* for the cluster c. Another annotator uses Google Goggles, an internet service that retrieves web pages that contain images similar to a given image. Google Goggles works well for distinctively textured images, logos, and texts. This annotator annotates object hypotheses with text (*text(c,txt)*) and logos (*logo(c,l)*). A brief description of the annotators used and their operation and results can be found in Table I.

An example of a pipeline of annotators is depicted in the upper part of Figure 2. The pipeline first tests the flatness of the individual object hypotheses. Subsequently, object hypotheses are annotated with a simple shape, the color, compared to known object models using the Linemod algorithm [2]. Finally, text and logo annotations are generated using the Google Goggles web service. A detailed performance analysis of the annotators will be presented in Section IV.

Given the annotations of objects AS, the probabilistic reasoning component of the perception system can be used to answer queries about any aspect of the respective probabilistic model Q. The probabilistic model is given by the joint probability distribution over the combination of the categories of objects and all possible annotations. The answer to the query is then the $argmax_Q P(Q \mid AS)$.

The learning process of the joint probability distribution over annotated scenes is depicted in the lower part of Figure 2. The learning of the the probability distribution and the probabilistic reasoning mechanisms are detailed in Section III.

III. Information Fusion

In the previous sections, we described how the raw sensory input data is being transformed into semantically more meaningful features by application of experts, the so-called annotators. However, since most of the annotators producing those object hypotheses are applied independently of each other, their outputs are not guaranteed to be globally consistent and they typically do not take into account object interactions in the current scene. In fact, their annotations might even be incorrect or contradictory. Therefore, in order to come up with a final ensemble decision, a strategy for combining all the annotations is needed.

To this end, we apply state-of-the-art methods from the field of statistical relational learning (SRL), a subfield of the machine learning discipline that has emerged and gained a lot of attraction in the recent couple of years. In SRL models, we can capture complex object interactions, represent and reason about object properties, their attributes and the relations that hold between them. Most notably, the ultimate strength of SRL models is their capability of allowing for reasoning about *all* observations simultaneously, taking into account interactions between objects and thus achieving a posterior belief that is guaranteed to be probabilistically sound and globally consistent.

In particular, we employ Markov Logic Networks (MLN) [8], a powerful knowledge representation formalism combining the expressive power of first-order logic (FOL) and the ability to deal with uncertainty of probabilistic

Prediction/Truth	Bowl	Cereal	Chips	Coffee	Cup	Fork	Juice	Ketchup	Knife	Milk	Mondamin	Oil	Pancake_maker	Pitcher	Plate	Popcorn	Pot	Salt	Spatula	Spoon	Toaster
Bowl	**10**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cereal	0	**8**	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Chips	0	0	**5**	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Coffee	0	0	0	**9**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cup	0	0	2	2	**20**	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
Fork	0	0	0	0	0	**1**	0	0	7	0	0	0	0	0	0	0	0	0	0	3	0
Juice	0	1	0	0	0	0	**12**	0	0	1	0	1	0	0	0	1	0	0	0	0	0
Ketchup	0	0	0	1	0	0	0	**7**	0	0	0	0	0	0	0	0	0	0	0	0	0
Knife	0	0	0	0	0	9	0	0	**6**	0	0	0	0	0	0	0	0	0	1	5	0
Milk	0	0	0	0	0	0	3	0	0	**10**	0	0	0	0	0	0	0	0	0	0	0
Mondamin	0	0	0	0	0	0	0	0	0	1	**7**	1	0	0	0	0	0	0	2	0	0
Oil	0	0	0	0	0	0	0	0	0	0	0	**7**	0	0	0	0	0	0	2	0	0
Pancake_maker	0	0	0	0	0	0	0	0	0	0	0	0	**6**	0	0	0	0	0	0	0	0
Pitcher	0	0	0	0	0	0	0	0	0	0	0	0	0	**3**	0	0	0	0	0	0	0
Plate	0	0	0	0	0	2	0	0	0	0	0	0	0	0	**23**	0	0	0	3	2	0
Popcorn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**5**	0	0	0	0	0
Pot	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**6**	1	1	0	0
Salt	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	**3**	0	0	0
Spatula	0	0	0	0	0	0	0	0	0	0	0	1	0	0	3	0	0	0	**10**	0	0
Spoon	0	0	0	0	0	3	0	0	6	0	0	0	0	0	0	0	0	0	2	**6**	0
Toaster	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	**4**

Fig. 3: Confusion Matrix for 10-fold cross-validation on the data set of 50 scenes. The rows represent the predictions, ground truth is given by the columns. Objects with the most severe confusion are knives, forks, spoons and spatulas.

graphical models. As opposed to most traditional machine learning approaches, learning and reasoning in MLNs is not restricted to a feature vector of fixed length, but is rather performed on whole databases of entities and relations.

Maintaining a joint probability distribution over observations, their class labels and the robot's current task context and belief state has several advantages over classical approaches and makes the systems' reasoning capabilities go far beyond traditional classifier systems:

a) collective classification: MLNs are able to simultaneously take into account any arbitrary but finite number of objects for classification. This an important feature for a perception system, since it captures interactions between objects in a scene. If a classification system is aware of the probability of jointly encountering two objects of particular types, this can tremendously boost the classification accuracy in real-world scenes. Encountering milk and cornflakes together on a table, for instance, is much more likely than finding cornflakes and ketchup.

b) confidence-rated output: Since the MLN for compiling annotations to a final decision is stacked upon the independent application of experts, such a probabilistic model is able to compensate for inconsistent annotations or uninformative features. If, for example, an annotator systematically confuses the shapes of clusters, the MLN will learn this erroneous hypotheses and treat them in a meaningful way.

c) generative models: An MLN representing a joint probability distribution can be used to infer answers to arbitrary queries about any aspect represented in the model. As our experiments will show, the MLN can also be used to reason about the most informative visual features when looking for a particular type of object in scene, for example.

d) ease of extension: integration of additional task-specific context information, or new specialized perception routines is straightforward. They just need to add their annotations to each of the object hypotheses and can be declaratively incorporated in the MLN.

A. Markov Logic Networks

More formally, an MLN consists of a set of formulas F in first-order logic and a real-valued weight w_i attached to each of those formulas F_i. The probability distribution over the set of possible worlds $\mathscr{X}$ represented by the MLN is defined as follows:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right), \tag{1}$$

where x is a complete truth assignment to all predicate groundings X (i.e. one possible world), $n_i(x)$ is the number of true groundings of formula F_i in x, and Z is normalization constant.

From a logical point of view, the outputs of the feature annotators can be regarded as tables in a relational database

and thus naturally correspond to predicates in FOL, and the segmented clusters represent the domain of discourse of entities we wish apply probabilistic, logical reasoning to. Furthermore, we can think of the final class label, i.e. the object category we wish to predict, as an additional predicate. As an example, consider a scene of two objects c_1 and c_2, where the ensemble of experts have identified c_1 being a yellow-ish box with a "Kellogg's" brand logo on it, and c_2 being a round, blue thing. We can capture such a scene in a relational database as follows:

$$\begin{aligned}
&\text{shape}(c_1, \textit{Box})\\
&\text{color}(c_1, \textit{Yellow})\\
&\text{logo}(c_1, \textit{"Kellogg's"})\\
&\text{color}(c_2, \textit{Blue})\\
&\text{shape}(c_2, \textit{Round})\\
&\text{category}(c_1, \textit{Cereal})\\
&\text{category}(c_2, \textit{Bowl}),
\end{aligned}$$

where we have manually added information about object classes in the "category" predicate. In MLNs, it is straightforward to create a model putting object attributes into relation with their class labels, since they provide a simple, declarative template language for generating probabilistic models. If we assume, for instance, that we can infer an object's category given its shape, a set of weighted formulas such as

$$\begin{aligned}
w_1 &= \log(0.66) \qquad \text{shape}(?x, \textit{Round}) \wedge \text{category}(?x, \textit{Bowl})\\
w_2 &= \log(0.33) \qquad \text{shape}(?x, \textit{Box}) \wedge \text{category}(?x, \textit{Bowl})
\end{aligned}$$

can be added to the model, which naturally represent the rules "everything is a round bowl" and "everything is a box-shaped bowl" (by default, all variables are universally quantified in MLNs). Of course, the above rules do not hold for most of the entities we encounter in the real world and, in fact, they can be considered mutually exclusive. However, according to (1), the probability distribution defined by this MLN indicates that any world in which we encounter a round bowl is twice as likely as a world in which we find a box-shaped bowl (assuming all other aspects being identical). Following this, we add such abstract, coarse "rules of thumb" to the MLN, modeling connections between the ensemble experts and the final ensemble decision. The weight parameters of the resulting MLN can be learned in a supervised learning manner.

IV. Experiments and Results

With the experiments presented in this section, we will prove the following properties of the proposed system. We will show that

1) hypotheses of individual annotators can be significantly boosted by applying SRL techniques,
2) it is beneficial to take into account object/object correlations in a perceptual classification model,
3) the proposed method is robust towards inconsistent annotations, which can be treated in a meaningful way,
4) our system's capabilities go far beyond traditional classifier systems, which are mainly given by discriminant functions with dedicated in- and output variables.

Annotator	# total Annotations	# correct Annotations	Predictive perf. (acc.)
Color	289	231 (79.9%)	17.5%
Goggles	80	—	21.2%
Prim. Shape	336	233 (69.3%)	26.1%
SACmodel	38	31 (81.5%)	
FlatObject	142	116 (81.6%)	
Linemod	90	45 (50%)	19.6%

Fig. 4: Evaluation of annotators in isolation: Correctness of their annotations and their predictive performance when applied in 10-fold cross-validation. Shape, SACmodel and FlatObject have been aggregated since they all contribute to the "shape" predicate.

Object	Accuracy	Precision	Recall	F_1-Score
Bowl	1.00	1.00	1.00	1.00
Cereal	0.98	0.80	0.80	0.80
Chips	0.99	0.83	0.71	0.77
Coffee	0.99	1.00	0.75	0.86
Cup	0.98	0.77	1.00	0.87
Fork	0.90	0.09	0.07	0.08
Juice	0.96	0.75	0.71	0.73
Knife	0.89	0.29	0.32	0.30
Ketchup	1.00	0.88	1.00	0.93
Milk	0.97	0.77	0.67	0.71
Mondamin	0.98	0.64	1.00	0.78
Oil	0.98	0.78	0.64	0.70
Pancake maker	1.00	1.00	1.00	1.00
Pitcher	1.00	1.00	1.00	1.00
Plate	0.95	0.77	0.82	0.79
Popcorn	0.99	0.83	0.83	0.83
Pot	0.99	0.75	1.00	0.86
Salt	0.99	0.75	0.75	0.75
Spatula	0.93	0.71	0.45	0.56
Spoon	0.91	0.35	0.38	0.36
Toaster	0.99	0.57	1.00	0.73

Fig. 5: Class-specific error measures for 10-fold cross-validation.

To this end, we arranged and recorded 50 realistic scenes, each comprising 5-10 instances of 21 different object categories, which can generally be found in typical kitchen scenarios. We discern four different kinds of scenarios: a breakfast table, a cooking scenario, a view into a refrigerator and a view into a kitchen drawer. The types of scenarios have been incorporated into each data set and can be regarded as task-specific knowledge about the current context of an activity. This is a reasonable presumption, since the location the robot is currently looking at can be assumed to be known from e.g. a map of the environment, and co-occurrences of objects are highly correlated in real-world scenarios. The object categories for each object have been labelled manually.

For the MLN we are using for obtaining a final ensemble decision of experts, we employ the logical predicates described in Table I, which naturally correspond to the annotator outputs in the system. Two additional predicates are used for specifying knowledge about the current context (i.e. the type of scenario) the perceptual task is performed in and for assigning a class label to each of the clusters in the scene at hand:

Ground Atom	Cereal	Chips	Cup	Pot
color(c,black)	0.3302	0.3476	0.2864	0.3582
color(c,blue)	0.2954	**0.3316**	0.2186	0.3148
color(c,red)	**0.3852**	**0.3656**	0.3452	0.3388
color(c,white)	0.3508	0.4216	0.2806	**0.3768**
color(c,yellow)	**0.4264**	**0.3484**	**0.4422**	0.2936
text(c,VITALIS_A)	**0.623**	0.0000	0.0000	0.0004
logo(c,DrOetker)	0.136	0.0006	0.0000	0.0000
logo(c,Kellogg's)	0.3734	0.0000	0.0000	0.0008
....	....	....	....	....
linemod(c,PfannerIce)	0.0004	0.0000	0.0008	0.0002
linemod(c,Popcorn)	**0.7392**	0.0006	0.0000	0.0010
linemod(c,Pot)	0.0008	0.0004	0.0004	**0.9994**
linemod(c,PringlesVin)	0.0000	0.0000	0.0004	0.0006
linemod(c,PringlesSalt)	0.0002	**0.4986**	0.0010	0.0006
....	....	....	....	....
shape(c,box)	**0.4806**	0.3870	0.2810	0.3556
shape(c,cylinder)	0.3722	**0.4540**	0.4010	0.4266
shape(c,flat)	0.3226	0.3682	0.2864	0.3862
shape(c,round)	0.3176	**0.4092**	**0.5182**	**0.4068**
size(c,big)	**0.368**	**0.3442**	**0.3768**	**0.3292**
size(c,small)	0.2626	0.2686	0.3148	0.2836

Fig. 6: (Partial) probabilities for different queries about visual features conditioned on the object class.

- scene($scene$): represents knowledge about the current context in which the perceptual task is being performed. possible contexts are dom($scene$) = $\{breakfast, cooking, drawer, fridge\}$
- category($cluster, object!$): assigns a class label to each cluster in the scene. In our experiments, we distinguished 21 different object categories (see also Figure 3).

In the MLN syntax, the "!" operator in a predicate declaration specifies that this predicate is to be treated as a functional constraint for the respective domain, i.e. for every cluster $c \in \text{dom}(cluster)$, there must be exactly one true ground atom among the ground atoms for the "category" predicate. In other words, we require exactly one object category association for each cluster. Since a particular cluster or entity cannot be of two different categories at a time, we argue that this model constraint is a reasonable assumption.

The following Markov Logic Network has been designed in order to model correlations between annotator outputs and the object classes:

$$w_1 \quad \text{size}(?c, +?sz) \wedge \text{shape}(?c, +?sp) \wedge \text{color}(?c, +?cl) \wedge \text{category}(?c, +?obj)$$
$$w_2 \quad \text{linemod}(?c, +?ld) \wedge \text{category}(?c, +?obj)$$
$$w_3 \quad \text{logo}(?c, +?logo) \wedge \text{category}(?c, +?obj)$$
$$w_4 \quad \text{text}(?c, +?text) \wedge \text{category}(?c, +?obj)$$
$$w_5 \quad \text{scene}(+?s) \wedge \text{category}(?c, +?obj)$$
$$w_6 \quad \text{category}(?c_1, +?t_1) \wedge category(?c_2, +?t_2) \wedge ?c_1 \neq ?c_2,$$

where the "+" operator specifies that the respective formula will be expanded to one individual formula for every value in the respective domain.

The domain "text" of the "text" predicate requires some special treatment: since its output is based on OCR text recognition by Google Goggles, this domain is potentially infinite and noise-prone. Thus, a mechanism is needed for transforming arbitrary strings into a proper set of symbolic constants. To this end, we applied a SAHN (sequential, agglomerative, hierarchical, non-overlapping) clustering technique to the values of the "text" domain in the training data before running the learning process. As a distance measure, we chose the well-known Levenshtein distance. Subsequently, every string (in both the training and test data) has been replaced by its nearest cluster centroid. This mechanism mitigates the negative effects of noise in the OCR annotations, since every unknown text is mapped to a known string which is explicitly represented in the model.

The weights have been determined by supervised learning of the manually labeled data using pseudo-log-likelihood learning with a Gaussian zero-mean prior of $\sigma = 10$, which serves regularization purposes.

For evaluating our model's performance in identifying object classes of entities in a scene given the observations from the annotators as evidence, we performed 10-fold cross-validation on the 50 scenes we recorded. The results are shown in Figure 3 and 5. As can be clearly seen, our model achieves reasonably high classification accuracies with respect to precision, recall and F_1-score. Indeed, the system achieves F $_1$-scores significantly above 70% for all objects except for the cutlery. The comparatively low performance on cutlery is to be expected, since the perceptual capabilities of the sensors and annotator experts are currently insufficient for distinguishing between the marginally observable differences between forks, spoons and knives.

However, we think that the overall performance of the system is remarkable compared to the individual performances of the single annotators in isolation. Figure 4 shows an evaluation or each of the annotators, counting the number of times an expert has annotated an object, the correctness of its annotations as well as the predictive performance an MLN consisting of only one formula containing the respective predicate would achieve. Note that Linemod's low performance is due to the fact that its main strenth is in recognizing untextured objects, but we created models for textured objects as well.

Figure 7 shows the confusion matrices for MLNs that have been trained with only one annotator each, in particular the goggles and the shape annotator. As can be seen, the individual annotators perform poorly on the entire data set, but each achieves quite good results in a particular subset. The goggles annotator, on the one hand, shows good performance on products and textured objects like the cereal boxes and the juice tetra-paks, but fails on cups and plates. On the other hand, the shape annotator fails on most of the products, but succeeds in identifying plates, cups and bowls. Hence, the single annotators can be regarded complementory with respect to their individual expertise, though neither of them is strong enough to perform well on all of the object classes. The ensemble given by the MLN, however, adapts to the individual strengths and weaknesses of the experts and thus can treat contradictory annotator outputs competently in order to come to a final decision.

Prediction/Truth	Bowl	Cereal	Chips	Coffee	Cup	Fork	Juice	Ketchup	Knife	Milk	Mondamin	Oil	Pancake_maker	Pitcher	Plate	Popcorn	Pot	Salt	Spatula	Spoon	Toaster
Bowl	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cereal	0	**8**	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
Chips	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coffee	0	0	0	**2**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cup	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fork	10	0	4	9	20	**15**	0	1	19	7	4	8	6	3	28	5	6	4	16	16	4
Juice	0	2	2	0	0	0	**14**	0	0	3	1	2	0	0	0	1	0	0	3	0	0
Ketchup	0	0	1	0	0	0	0	**5**	0	0	0	0	0	0	0	0	0	0	0	0	0
Knife	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
Milk	0	0	0	0	0	0	0	0	0	**3**	0	0	0	0	0	0	0	0	0	0	0
Mondamin	0	0	0	0	0	0	1	1	0	1	**1**	0	0	0	0	0	0	0	0	0	0
Oil	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0
Pancake_maker	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
Pitcher	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0
Plate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0
Popcorn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0
Pot	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0
Salt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
Spatula	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	**3**	0	0
Spoon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
Toaster	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

(a) 10-fold cross-validation using only the Goggles annotator.

Prediction/Truth	Bowl	Cereal	Chips	Coffee	Cup	Fork	Juice	Ketchup	Knife	Milk	Mondamin	Oil	Pancake_maker	Pitcher	Plate	Popcorn	Pot	Salt	Spatula	Spoon	Toaster
Bowl	**10**	0	0	3	6	0	0	4	0	2	2	4	0	3	4	0	5	3	13	0	0
Cereal	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chips	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Coffee	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cup	0	0	7	0	**14**	0	0	3	0	0	5	7	0	0	0	0	0	0	1	0	0
Fork	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Juice	0	10	0	9	0	0	**17**	0	0	13	0	0	6	0	0	6	1	1	1	0	4
Ketchup	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0
Knife	0	0	0	0	0	11	0	0	**12**	0	0	0	0	0	13	0	0	0	2	9	0
Milk	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0
Mondamin	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0
Oil	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0	0
Pancake_maker	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
Pitcher	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0
Plate	0	0	0	0	0	4	0	0	7	0	0	0	0	0	**11**	0	0	0	5	7	0
Popcorn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0
Pot	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0
Salt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
Spatula	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0
Spoon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0
Toaster	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**

(b) 10-fold cross-validation using only the shape annotator.

Fig. 7: Evaluation of 2 of the annotators in isolation, i.e. just considering them in the MLN. The results show that the different annotators complement each other and compensate for the errors of the other.

Inferring the most probable categories given the observed properties of each objects is only one possibile kind of queries the system can answer. Indeed, the learned joint probability distribution on objects and their attributes allows reasoning about *arbitrary* queries with respect to any variable that is contained in the model. Our approach can also be used to reason about the perceptual features to be expected when looking for a particular object in a scene. If the robot is supposed to find a box of cereals on a breakfast table, for instance, the following query can be formulated in order to retrieve the most informative features for distinguishing the cereal box from other objects:

$$P\left(\begin{array}{l|l} \text{shape}(c,?sh),\text{color}(c,?c), & \text{scene}(\textit{breakfast}), \\ \text{size}(c,?s),\text{logo}(c,l?),\text{text}(c,?t) & \text{category}(c,\textit{Cereal}) \end{array}\right).$$

Figure 6 shows an excerpt of the probability distribution computed for the above query, where the most probable solution is printed bold. According to the most probable solution, we can deduce symbolic descriptions of expected visual properties of different kinds of objects: The cereals are expected to be a big, yellow and red box, on which we can read the text "VITALIS _A", and the linemod annotator would consider it popcorn (which is one more example of how the MLN compensates for the errors of individual experts).

V. Related Work

Most autonomous manipulation robots employ perception systems that are trained with appearance models of the objects they are to detect, recognize, and localize. In operation, the robot uses a database of trained objects to match them against the perceived sensor data. Successful examples of such robot object perception systems are described in [9] using point features of 3D opaque objects, or *MOPED*, which uses visual keypoint descriptors for the learned textured objects [10] and specialized perception systems for translucent objects [11]. In the presented work we encapsulate these methods as annotators, use them as experts, and therefore boost the performance by exploiting these algorithms.

A variety of methods exist that can handle reasonably well some of the subproblems of perception. Many of these methods are complementary and could be combined to enhance performance. Examples of such methods are door handle detectors [12]. Again, such methods are to be included in the future as parts of our system.

With respect to its operation, our approach falls into the category of unstructured information management systems – systems that look for segments of unstructured information that have a deeper structure. In 3D perception, RGB-D point clouds can be considered as unstructured information that contain object hypotheses as nuggets of more structured information. Object hypotheses have several perceptual features as well as symmetry and compactness properties. Unstructured information processing and management primarily facilitates hypothesis generation, testing, and ranking and the use of ensembles of expert methods. Unstructured information management is primarily investigated in the area of webscale information systems, most prominently in the context of the Watson system [13]. In our research, we transfer and modify this technology for its use in robot perception.

Ensemble of expert-based systems have proven to be very successful in the area of machine learning [14] and hold great promise for boosting the perceptual capabilities of robots. Polikar [15] presents a thorough analysis of ensemble based systems, giving several reasons in favor of choosing them (e.g. statistical, lack or abundance of data, data fusion).

An example of a robotic perception system employing the ensemble of experts idea is presented by Okada *et. al.* [16]. Multiple detectors and views were combined based on particle filters. The probabilistic fusion of different results corresponds to a simple rule ensemble, i.e. one that is not trainable.

In recent years attribute based perception has received a lot of attention. In the context of robotics Sun *et. al.* [17]

introduce the combination of appearance attributes and object names in order to identify objects in a scene. Pronobis *et. al.* [**?**] describe a framework for semantic mapping based on a combination of object attributes, room appearance and human input. Our approach is more like a combination of the two approaches, having as inputs only visual cues of the objects and the domain knowledge (scene type).

VI. Conclusions

In this work, we propose a novel perception system for robots acting in everyday human environments. In contrast to existing systems, which mainly focus on employing a specific perceptual algorithm, our approach follows the paradigm of ensembles of experts – sets of diverse and highly specialized algorithms that are strategically combined in order to draw a well-informed final conclusion.

Ensemble-based systems like the AdaBoost algorithm [14] have been used in previous works, but traditional machine learning methods suffer from their inability to collectively classify arbitrary numbers of objects in a scene. The proposed system instead makes use Markov Logic Networks, a powerful relational probabilistic framwork allowing to take into consideration also object-interactions like they are encountered very frequently in real-world scenarios. We prove the strength of our approach by a profound evaluation of our system's performance on a data set 50 typical kitchen scenarios. We show that the SRL techniques employed are well-suited for the computation of a posterior belief for a given query, though the expressiveness of the individual annotators is quite poor.

In the future, we plan to integrate more and better algorithms (e.g. MOPED for textured objects) in order to have more experts making decision about the same domain. Studying how taking into account the confidences of these experts could improve our system seems as the next most important step to take.

Given that in a task specific scenario we can query for the most descriptive features of an object, using the resulting information will enable the systems to choose the feature detectors that are most appropriate, and use these for detection/tracking. Doing so we hope to achieve better execution times for our perception system.

An interesting future endavour will be to try our system during robot operation, again making use of the fact that we can formulate arbitrary queries about any variable that is contained in the model. For example, given the robots location in the environment, and the type of object we are looking for we need to decide upon which actions to take to find the object as fast as possible (e.g. move closer, query the *memory* of the robot, or call specific perception routines).

We argue that the use of ensemble-based systems and specialized perception routines is a key paradigm for pushing the perceptual capabilities of our today's robots to more versatile and advanced applications.

Acknowledgement

This work is supported in part by the EU FP7 projects *RoboHow* (grant number 288533) and *ACAT*(grant number 600578).

References

[1] L. C. Goron, Z. C. Marton, G. Lazea, and M. Beetz, "Segmenting cylindrical and box-like objects in cluttered 3D scenes," in *7th German Conference on Robotics (ROBOTIK)*, Munich, Germany, May 2012.

[2] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[3] D. Pangercic, M. Tenorth, B. Pitzer, and M. Beetz, "Semantic object maps for robotic housework - representation, acquisition and use," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, October, 7–12 2012.

[4] A. Richtsfeld, T. Morwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of unknown objects in indoor environments," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 4791–4796.

[5] A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1584–1601, October 2006.

[6] Z.-C. Marton, F. Balint-Benczedi, F. Seidel, L. C. Goron, and M. Beetz, "Object Categorization in Clutter using Additive Features and Hashing of Part-graph Descriptors," in *Proceedings of Spatial Cognition (SC)*, Abbey Kloster Seeon, Germany, 2012.

[7] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011, pp. 1–4.

[8] M. Richardson and P. Domingos, "Markov Logic Networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[9] A. Aldoma, Z.-C. 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," *Robotics & Automation Magazine*, vol. 19, no. 3, pp. 80–91, September 2012.

[10] A. Collet Romea, M. Martinez Torres, and S. Srinivasa, "The MOPED framework: Object recognition and pose estimation for manipulation," *International Journal of Robotics Research*, vol. 30, no. 10, pp. 1284 – 1306, September 2011.

[11] I. Lysenkov, V. Eruhimov, and G. Bradski, "Recognition and Pose Estimation of Rigid Transparent Objects with a Kinect Sensor," in *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[12] T. Rühr, J. Sturm, D. Pangercic, D. Cremers, and M. Beetz, "A generalized framework for opening doors and drawers in kitchen environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, May 14–18 2012.

[13] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty, "Building Watson: An overview of the DeepQA project," *AI Magazine*, vol. 31, no. 3, pp. 59–79, 2010. [Online]. Available: http://www.aaai.org/ojs/index.php/aimagazine/article/view/2303

[14] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, 1996, pp. 148–156.

[15] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[16] K. Okada, M. Kojima, S. Tokutsu, T. Maki, Y. Mori, and M. Inaba, "Multi-cue 3D object recognition in knowledge-based vision-guided humanoid robot system," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*, pp. 3217–3222, 2007.

[17] Y. Sun, L. Bo, and D. Fox, "Attribute Based Object Identification," in *IEEE International Conference on on Robotics and Automation*, 2013.