Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Exploiting Semantic Knowledge for Robot Object Recognition

José-Raúl Ruiz-Sarmiento[a,*], Cipriano Galindo[a], Javier Gonzalez-Jimenez[a]

[a]*System Engineering and Automation Dept., University of Málaga, Campus de Teatinos, 29071, Málaga, Spain.*

Abstract

This paper presents a novel approach that exploits semantic knowledge to enhance the object recognition capability of autonomous robots. Semantic knowledge is a rich source of information, naturally gathered from humans (elicitation), which can encode both objects' geometrical/appearance properties and contextual relations. This kind of information can be exploited in a variety of robotics skills, especially for robots performing in human environments. In this paper we propose the use of semantic knowledge to eliminate the need of collecting large datasets for the training stages required in typical recognition approaches. Concretely, semantic knowledge encoded in an ontology is used to synthetically and effortless generate an arbitrary number of training samples for tuning Probabilistic Graphical Models (PGMs). We then employ these PGMs to classify patches extracted from 3D point clouds gathered from office environments within the UMA-offices dataset, achieving a $\sim 90\%$ of recognition success, and from office and home scenes within the NYU2 dataset, yielding a success of $\sim 81\%$ and $\sim 69.5\%$ respectively. Additionally, a comparison with state-of-the-art recognition methods also based on graphical models has been carried out, revealing that our semantic-based training approach can compete with, and even outperform, those trained with a considerable number of real samples.

Keywords:
Semantic Knowledge, Human Elicitation, Object Recognition, Probabilistic

*Corresponding author
Email addresses: jotaraul@uma.es (José-Raúl Ruiz-Sarmiento), cipriano@ctima.uma.es (Cipriano Galindo), javiergonzalez@uma.es (Javier Gonzalez-Jimenez)

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Graphical Models, Autonomous Robots

1. Introduction

Object recognition is one of the key abilities of a mobile robot intended to perform high-level tasks in human environments, where objects are usually placed according to their functionality, e.g., tv-sets are in front of couches, night tables are near beds, etc. As reported by other authors (Galleguillos and Belongie, 2010), the exploitation of these contextual relations, that can be seen as a form of *semantic knowledge*, can improve the performance of traditional object recognition methods which only rely on sensorial features.

To illustrate the benefits of using semantics, let's consider a robot coping with the task of recognizing the objects placed in its surroundings. This may become complex for a number of reasons, including the large number of possible object classes and features to extract, their similarity, etc. Suppose now that the robot knows that it is in an office and has some semantic knowledge related to that particular domain, for example the type of objects usually present in a typical office environment and their contextual relations. This simplifies the recognition problem, drastically reducing the range of possible objects classes, and even more importantly, enabling the recognition system to exploit particular object relations to gain in effectiveness and robustness. For instance, an object that resembles an office table according to its geometry can be more confidently recognized as such if objects typically found near it, e.g. a computer screen and/or a chair, are also detected and fulfill certain contextual relations, for example, the computer screen is on the table and the chair is close to it.

In this work we present a novel approach that exploits semantic knowledge encoded by *human elicitation* to train *Probabilistic Graphical Models* (PGMs) (Koller and Friedman, 2009) for object recognition. PGMs form a machine learning framework that is widely applied to object recognition given its capabilities for modelling both uncertainty and objects relations. These systems need a vast amount of training data in order to reliably encode the gist of the domain at hand, however, the gathering of that information is an arduous, time-consuming, and – in some domains – not a tractable task. To face this issue, we codify semantic knowledge by means of an ontology (Uschold and Gruninger, 1996), which defines the domain object classes, their properties, and their relations, and use it to generate training samples

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

for a Conditional Random Field (CRF) (Koller and Friedman, 2009). These training samples reify prototypal scenarios where objects are represented by a set of geometric primitives, e.g., planar patches or bounding boxes, that fulfill certain geometric properties and relations, like proximity, difference of orientation, etc.

Aiming to show the performance of CRFs trained with the proposed approach, they have been integrated into an object recognition framework. This framework operates by processing point clouds provided by a RGB-D camera, in order to extract geometric primitives (see figure 1-a), which are then recognized as belonging to a certain object class through an inference process over the trained CRF. We have obtained promising results in office and home environments, employing both planar patches and bounding boxes as geometric primitives, though our methodology can be applied to other scenarios and sensorial data types.

In the literature, PGMs are used, in general, to learn the properties of the different object classes and their contextual relations using data from previously collected datasets. In contrast, the work presented here drives this learning phase by providing synthetic training samples extracted from the semantic knowledge of the domain at hand. This knowledge can be naturally provided by humans and encoded into an ontology, and exhibits three advantages with respect to other related approaches:

- It eliminates the usually complex and high resource-consuming task of collecting the large number of training samples required to tune an accurate and comprehensive model of the domain.

- Ontologies are compact and human-readable knowledge representations. In that way, extending the problem with additional object classes is just reduced to codify the knowledge about the new classes into the ontology, generate synthetic samples considering the updated semantic information, and train the CRF. This process can be completed in a few minutes, in contrast to the time needed for gathering and processing real data.

- The recognized objects are anchored to semantically defined concepts, which is useful for robot high-level tasks like reasoning or task planning (Galindo and Saffiotti, 2013; Galindo et al., 2007; Coradeschi and Saffiotti, 2003).

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

We have conducted an evaluation of our work employing two datasets: one from our facilities, called UMA-offices, which counts 25 office environments, and the NYU2 dataset (Silberman et al., 2012), from which we have extracted 61 offices and 200 home scenes. The performance of CRFs trained with our methodology have been also compared with two state-of-the-art methods, namely i) a standard formulation of CRFs trained and tested with real data (Koller and Friedman, 2009), and ii) the CRF presented in Xiong and Huber (2010). The results show that our approach can compete with, and even outperform, those trained with a considerable number of real samples.

In the next section we put our proposal in the context of other related works. Section 3 introduces probabilistic graphical models applied to object recognition, while in section 4 we present the proposed method to train these models using semantic knowledge. In section 5, the evaluation results of the method considering two datasets comprising office and home environments are shown, and a comparison with other state-of-the-art approaches is presented. Finally, section 6 ends with some conclusions and future work.

2. Related work

Object recognition is a key topic in robotics and computer vision that, in many cases, has been successfully addressed by *only* using the visual features of isolated objects, i.e. without considering information from the rest of the scene. Some remarkable examples are the Viola and Jones boosted cascade of classifiers (Viola and Jones, 2001), the SIFT object recognition algorithm (Lowe, 2004) or the Bag of Features (Nister and Stewenius, 2006) models. However, the current trend also considers the exploitation of contextual information between objects, aiming to improve the recognition results (see Galleguillos and Belongie (2010)).

Throughout this section, we discuss related works on object recognition systems that resort to graphical models or semantic knowledge to model contextual information. Also, some works reporting different alternatives to the utilization of ontologies as a source of semantic information for object recognition are commented.

2.1. Probabilistic Graphical Models

Probabilistic Graphical Models (PGMs) (Koller and Friedman, 2009) is one of the most resorted frameworks to manage contextual information. The

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

earliest works using this tool for object recognition are based on intensity information of the scene, like (Xiang et al., 2010), where the context between pixels in a given RGB image is modelled by a discriminative Conditional Random Field (CRF). Another work also relying on intensity images is the presented in Quattoni et al. (2004) that proposes a CRF framework which incorporates hidden variables for part-based object recognition. The work in Mottaghi et al. (2011) also builds part-based models of objects, and represents their interrelations with a PGM. More recent is the work presented in Floros and Leibe (2012) which employs stereo intensity images in a CRF formulation. Three-dimensional information from stereo enables the exploitation of meaningful geometric properties of objects and relations. However, stereo systems are unable to perform on surfaces/objects showing an uniform intensity, which can negatively affect the recognition performance.

With the emergence of inexpensive 3D sensors, like Kinect, a new batch of approaches have appeared leveraging the dense and relatively accurate data provided by these devices. For example, the work presented in Anand et al. (2013) builds a model isomorphic to a Markov Random Field (MRF) according to the segmented regions from a scene point cloud and their relations. The authors did the tedious work of gathering information from 24 office and 28 home environments, and manually labelled the different object classes. Interestingly, it is shown in Ren et al. (2012) that the accuracy of a MRF in charge of assigning object classes to a set of superpixels increases as the amount of available training data augments. In Valentin et al. (2013) a meshed representation of the scene is built on the basis of a number of depth estimates, and a CRF is defined to classify mesh faces. CRFs are also used in Kahler and Reid (2013) and Xiong and Huber (2010), where Decision Tree Fields (Nowozin et al., 2011) and Regression Tree Fields (Jancsary et al., 2012) are studied as a source of potentials for the PGM. The CRF structure for representing the scenes in Xiong and Huber (2010) is similar to the one presented here. In that work, a CRF is used to classify the main components of a facility, namely clutters, walls, floors and ceilings.

All the methods mentioned above require the collection of large datasets that adequately capture the variability of the domain, which can be a tedious, repetitive, and time-consuming task that consists of moving the robot from one scene to another, gathering the data, and post-processing it accordingly to the type of information expected by the training algorithms. The claim of this work is the utilization of semantic knowledge codified into an ontology as a valuable source of information for the generation of synthetic

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

training samples that, being representative of the domain, also can capture its variability.

2.2. Semantic Knowledge

In the literature, some alternatives to PGMs for object context modelling have been also reported. For example, in Günther et al. (2013) a system relying on an ontology plus rules defined into the Semantic Web Rule Language (Horrocks et al., 2004) is used to generate object hypotheses. These hypotheses are subsequently checked in a matching process with CAD models. Another example is Nüchter and Hertzberg (2008), where a constraint network implemented in Prolog classifies the main structural surfaces, i.e. walls, floors, ceilings and doors, using contextual relations like orthogonal, parallel, above, etc. Nevertheless, these methodologies are unable to handle uncertainty, and exhibit difficulties to leverage all the potential of the contextual relations.

2.3. Alternative sources of information

Additionally to the use of semantic knowledge, other sources of information can be also considered to codify and manage the knowledge from a given domain. For example, in Zhou et al. (2012), a web mining knowledge acquisition system is presented as a mechanism to obtain information about the location of objects. In Fergus et al. (2005) the authors describe PGMs that are trained with images from the Google's image search engine. They reported that the high percentage of low quality search results (e.g images where the object of interest appears occluded or is missing, cartoons instead of real objects, etc.) represents a serious drop in the recognition performance. Knowledge bases, like ConceptNet (Speer and Havasi, 2013), and language models, like TypeDM (Baroni and Lenci, 2010), have been also studied for visual recognition tasks in Le et al. (2013), concluding that they can be inconsistent with the expectation of the presence of objects in the real world if insufficient objects and/or relations are included. Another example of exploitation of encoded information about objects' relations is Kunze et al. (2014), where the search of a given object is directed by a previously learnt Gaussian Mixture Model (GMM).

In comparison with those methods, the codification of the domain knowledge through human elicitation as presented in this work enables a truly and effortless encoding of a large number of objects' features and relations between them. Moreover, since the source of semantic information (a person or

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

a group of people) is trustworthy, in contrast to online search or web mining-engine based methodologies, there is less uncertainty about the validity of the information being managed. This enables the use of such a semantic information for generating training data which is well representative of the domain. In addition, the use of an ontology to structure that knowledge permits the robot to take advantage of it for other high level applications (Galindo et al., 2008; Galindo and Saffiotti, 2013).

3. Scene Object Recognition through Conditional Random Fields

Conditional Random Fields (CRF) Koller and Friedman (2009) are a particular case of Probabilistic Graphical Models that relies on conditional probability distributions. When applied to object recognition, a CRF computes the posterior $P(\mathbf{y}|\mathbf{x})$, where $\mathbf{x} = [x_1, x_2, ..., x_n]$ are observations of elements in the scene, and $\mathbf{y} = [y_1, y_2, \ldots, y_n]$ are random variables representing the classes of these elements from the set L of the possible object classes. Figure 1-b shows an example where $L = \{computer\_screen, table, chair\_back, chair\_rest, floor, wall\}$.

The posterior $P(\mathbf{y}|\mathbf{x})$ can be calculated by computing the probability of each possible assignation to the variables in $\mathbf{y}$ conditioned to $\mathbf{x}$, which can become unfeasible if the number of possible assignations is high. CRFs overcome this issue by compactly encoding $P(\mathbf{y}|\mathbf{x})$ through a graph structure that captures the dependence relations among random variables. Concretely, a CRF factorizes $P(\mathbf{y}|\mathbf{x})$ over an undirected graph $H = (V, E)$, where V is a set of nodes, one per each random variable in $\mathbf{y}$, and E is the set of edges linking nodes that are contextually related. These relations are established according to the semantics of the domain and the geometry of the scene. For example, in the CRF structure of figure 1-c defined from the observations in figure 1-a, the nodes y_3 and y_5 are linked due to the proximity in the scene of their related observed planar patches ID_3 and ID_5. The intuition behind this is that only the neighbors of an object will directly influence its recognition, as stated by the Markov properties (Koller and Friedman, 2009).

According to the Hammersley-Clifford theorem (Koller and Friedman, 2009), the factorization of $P(\mathbf{y}|\mathbf{x})$ over a CRF can be expressed as a product of factors. A factor is a function associated to a random variable or a set of variables that represents a probability distribution over it/them. In this work we consider two types of factors: *unary* and *pairwise* (see figure 1-c). Unary factors encode knowledge about the properties of the object itself and

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

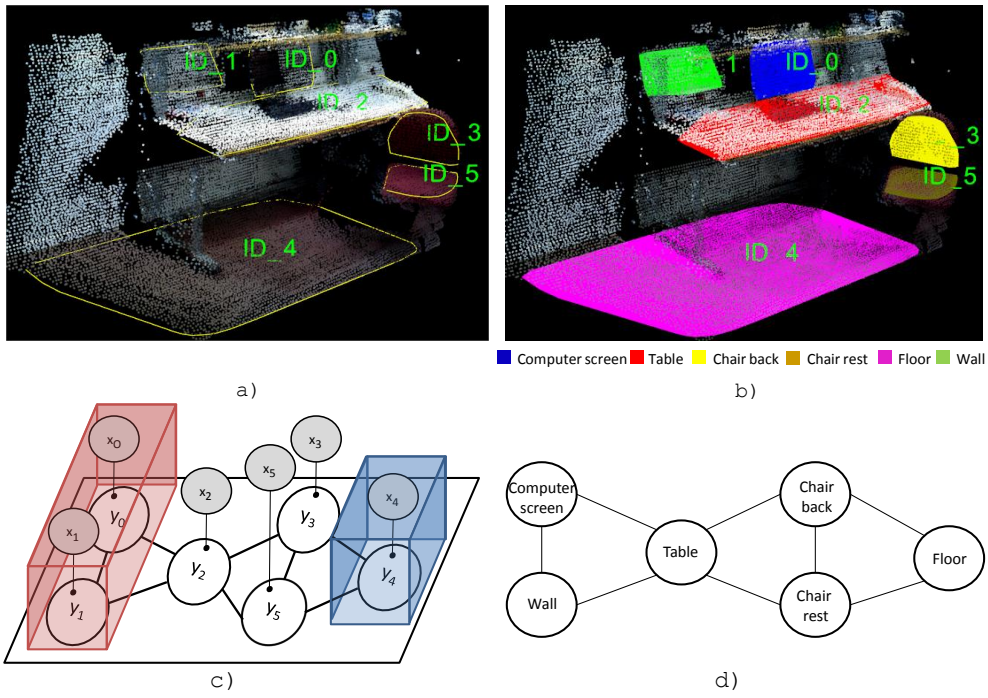

Figure 1: a) Example of a scene segmented into planar patches (labeled with an ID and delimited by yellow lines). b) Scene objects recognized by our method. c) Graphical model built for the planar patches shown in a). Each patch is associated to a node $y_0, .., y_5$, whose probabilistic distributions are conditioned to their respective patch observations $x_0, .., x_5$ (observation x_i corresponds to patch ID_i). Near patches are linked by an edge. The blue box encapsules the scope of a particular unary factor, while the red one shows the scope of a pairwise factor. d) The resultant graphical model after the execution of the recognition method, when random variables take a value according to their most probable assignation.

therefore affect to single nodes. On the other hand, pairwise factors act over connected variables, and encapsulate knowledge about the objects' relations. In other words, unary factors model how likely an object y_i belongs to a certain class in L based only on the observed properties x_i, whereas pairwise factors state the compatibility of an object assignation with respect to the classes of its neighboring objects.

More concretely, we define an unary factor, denoted by $U(\cdot)$, as a linear model:

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

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Table 1: Unary and pairwise features used in this work to characterize planar patches of the scene.

id	**Unary features**
l_1	Centroid height from the floor.
l_2	Orientation w.r.t. the horizontal.
l_3	Area of its bounding box.
l_4	Elongation.
id	**Pairwise features**
i_1	Perpendicularity.
i_2	on/under relation.
i_3	Vertical distance of centroids.
i_4	Ratio between areas.
i_5	Ratio between elongations.

where $f(x_i)$ computes a vector of features that characterizes the object x_i, $\boldsymbol{\omega}_l$ is a vector of weights for the class l obtained during the training phase, and $\delta(y_i = l)$ is the Kronecker delta function, which takes value 1 when $y_i = l$ and 0 otherwise. Table 1-top shows the unary features used in this work. As an example, let's consider the planar patch *ID_0* representing a computer screen in figure 1, which corresponds to observation x_0. In this case, the outcome of the $f(\cdot)$ function is $f(x_0) = [1.06, 0, 0.17, 1.83]$, where 1.06 stands for its centroid height, 0 its orientation, and so on.

On the other hand, we define the pairwise factor $I(\cdot)$ as:

$$I(y_i, y_j, x_i, x_j, \boldsymbol{\theta}) = \sum_{l1 \in L} \sum_{l2 \in L} \delta(y_i = l_1, y_j = l_2) \boldsymbol{\theta}_{l_1 l_2} g(x_i, x_j) \tag{2}$$

where the function $g(x_i, x_j)$ computes pairwise features between the observations x_i and x_j, and $\boldsymbol{\theta}_{l_1 l_2}$ is a vector of weights for the pair of classes l_1 and l_2. Table 1-bottom enumerates the pairwise features used to characterize the objects' relations.

For convenience, the product of factors over the posterior probability P can be expressed by means of log-linear models as:

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

where $Z(\cdot)$ is the normalizing partition function so $\sum_{\xi(\mathbf{y})} p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}, \boldsymbol{\theta}) = 1$, being $\xi(\mathbf{y})$ an assignation to the variables in $\mathbf{y}$, and $\epsilon(\cdot)$ the so-called energy function defined as:

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

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

3.1. Training the Model

Training a CRF consists of estimating the vectors of weights $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ that maximize the likelihood function:

$$\max_{\boldsymbol{\omega}, \boldsymbol{\theta}} L_P(\boldsymbol{\omega}, \boldsymbol{\theta} | D) = \max_{\boldsymbol{\omega}, \boldsymbol{\theta}} \prod_{d \in D} P(\mathbf{y}_d | \mathbf{x}_d, \boldsymbol{\omega}, \boldsymbol{\theta}) \tag{5}$$

where $D = \{d_1, d_2, ..., d_m\}$ is a dataset composed of m training samples. Each training sample contains the observations to be recognized $\mathbf{x}_d$ labeled with their ground truth object classes in $\mathbf{y}_d$. Solving equation 5 requires the calculation of the partition function Z, which becomes computationally intractable in practice. To overcome this problem, it is common to resort to the pseudo-likellihood, instead (Koller and Friedman, 2009). It consist of an alternative, tractable objective function for which the estimation of $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ converges to those computed by the likelihood one if a sufficient large number of samples is provided.

As commented, the training dataset must be comprehensively enough to accurately capture the characteristics and variability of the domain. At this point, the exploitation of semantic knowledge brings two interesting advantages: (i) it provides synthetic training samples that naturally encode the variability of the domain (as it is shown in section 4.2), and (ii) it eliminates the task of gathering, processing and labelling sensorial data to generate a sufficiently comprehensive dataset.

3.2. Inference

Given the observation of a scene, the graph $H = (V, E)$ is built according to the sensed elements $\mathbf{x}$ and the conditional dependencies between the random variables $\mathbf{y}$, as described above. Thereby, the recognition problem consists of finding the assignation to the variables in $\mathbf{y}$ that maximizes the posterior, that is:

$$\begin{aligned} \hat{y} &= arg \max_{y} P(\mathbf{y} | \mathbf{x}, \boldsymbol{\omega}, \boldsymbol{\theta}) \\ &= arg \max_{y} \frac{1}{Z(\mathbf{x}, \boldsymbol{\omega}, \boldsymbol{\theta})} e^{-\epsilon(\mathbf{y}, \mathbf{x}, \boldsymbol{\omega}, \boldsymbol{\theta})} \end{aligned} \tag{6}$$

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

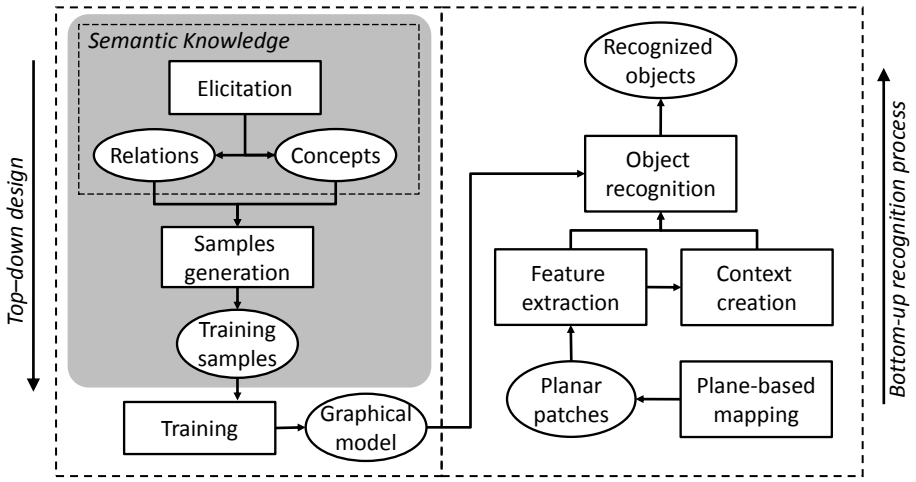

Figure 2: Overview of the developed framework for object recognition. The shadowed area delimitates the proposed components for the generation of training samples. Boxes represent processes, whereas ovals are generated/consumed data.

Since the partition function does not depend on the assignations to $\mathbf{y}$, we can simplify this expression to:

$$\hat{y} = arg\max_{y} e^{-\epsilon(\mathbf{y}, \mathbf{x}, \boldsymbol{\omega}, \boldsymbol{\theta})} \tag{7}$$

This equation is known as the Maximun a Posteriori (MAP) query or Most Probable Explanation (MPE). Although we avoid the computation of the partition function, the exact computation of this query is still unfeasible, as the number of possible configurations is exponential with the number of nodes in V. To overcome this issue, we use the Iterated Conditional Modes (ICM) algorithm (Besag, 1986).

As an illustrative example, figure 1-d displays the values taken by the nodes of the graph in figure 1-c after the inference process, and figure 1-b shows these results in the scene.

4. Using Semantic Knowledge for Training

The proposed method for training PGMs according to semantic knowledge follows a *top-down* methodology (see figure 2). The design starts with the definition of an ontology for the knowledge domain at hand, e.g. an office

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

environment, through human elicitation, stating the typical objects, their geometrical features, and relations. Then, the encoded semantic knowledge is used for generating sets of synthetic samples, which replace the real datasets required for training.

Once the PGM is trained, and aiming to show its performance, it is integrated into an object recognition framework that works following a *bottom-up* stance (see figure 2). During the robot operation, a plane-based mapping algorithm (Fernandez-Moral et al., 2013) extracts planar patches, which are characterized through a number of features, e.g., size, orientation, position or contextual relations. These characterized planar patches feed the inference process described in section 3.2.

The next section details the process for encoding the semantic knowledge provided by human elicitation into an ontology, and then section 4.2 describes its utilization for generating an arbitrary number of synthetic training samples.

4.1. Ontology Definition through Human Elicitation

An ontology is a representation of a conceptualization related to a knowledge domain that consists of a number of *concepts* arranged hierarchically, *relations* among them, and *instances* of concepts, also called *individuals* (Uschold and Gruninger, 1996). For example, an office environment can be represented by an ontology of concepts defining rooms and objects, e.g. `meeting_room`, `office_table` or `printer`, and instantiations of such concepts, e.g. `meeting_room-1`, which refers to a particular meeting room. Ontologies also comprise relations among concepts like ‘‘`Object has_location Room`’’, which establishes that the instances of the concept `Object` are (can be) located at a particular instance of `Room`. For instance, a possible relation can be ‘‘`office_table-2 has_location meeting_room-1`’’. The ontologies used in this work are defined by human elicitation, a process that enables the exploitation of its experience and knowledge[1] for setting the features and relations among the domain concepts.

Figure 3-a) depicts part of the office ontology defined in our experiments. The root concept is `Object`, with three subconcepts: `Device`, `Furniture` and `Building`, which represents the objects that are typically found in office

[1]Please notice that the source of this information could be also a large number of humans, i.e. crowd-sourcing.

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

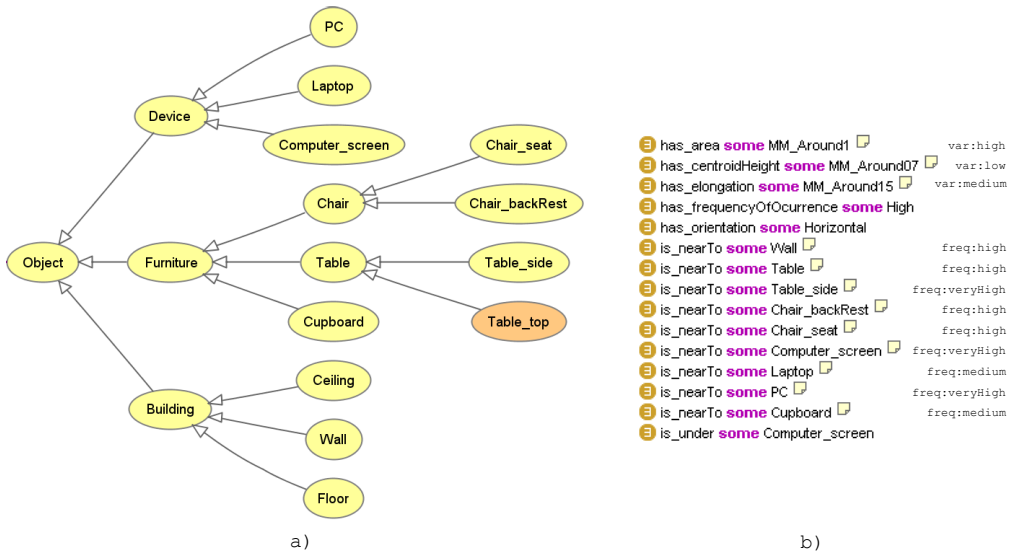

Figure 3: a) Hierarchy of concepts defined in the office ontology used in this work. b) Definition of the `Table_top` concept based on properties, relations and annotations.

environments. Notice that the person can vary the granularity of the defined concepts, as it is the case of the concept `Table` that has been split into two related concepts: `Table_top` and `Table_side`.

The geometrical properties considered by the human to describe these concepts and their relations are enumerated in table 2. Such properties can be interpreted as restrictions to be fulfilled by instances belonging to that concepts. Additionally, they compound the minimum set of properties that permits a human to distinguish between the object classes employed during the method evaluation (see section 5). For example, figure 3-b) shows the definition of the concept `Table_top`, restricting the geometric features and relations considered for a standard table top.

The geometric features defined over the concepts are useful to describe the typical shape, size or relative position of their instances. However, not all the instances of a particular concept have exactly the same appearance in the real world. To quantify objects' variability, the person may also annotate the encoded restrictions with a discrete value from the set $R_A = \{null,\ veryLow,\ low,\ medium, high, veryHigh\}$. Thus, according to the `Table_top` definition given in 3-b), its height shows a *low* variability around the established value of $0.7m$, indicating that most tables share this

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Table 2: Properties defined into the ontology.

Name	Meaning
has_area	Area of the object in m^2.
has_centroidHeight	Height of the object centroid w.r.t. the floor in m.
has_elongation	Ratio between the object length in its two main directions.
has_frequencyOfOccurrence	How often an object appears in the studied environment.
has_orientation	Main orientation of the object.
is_nearTo	An object is near to other one.
is_on	An object is placed on another one.
is_under	An object is placed under another one.

typical height. The area, however, can largely vary from the averaged value, i.e. $1m^2$, expressing the differences in size of the tables that can be found in an office. Given that the same set of geometric features is employed for describing all the concepts during the elicitation process, the time needed for their definition scales linearly with the number of object classes. It is also worth to mention that, although the definition of the objects' variability by means of elements of the set R_A could seem subjective (i.e. dependant on the person): the objectiveness can be increased through crowd-sourcing; the crispy values from R_A are relevant but not determinant during the generation of synthetic data – see section 4.2.

Proximity restrictions between objects are also incorporated into the ontology with a value from the R_A set, but with a different meaning. In this case, it is indicated how frequently a particular relation holds. For instance, the person establishes that a computer screen and a table top likely appear close to each other by adding an annotation with the value *veryHigh* (see figure 3-right). Note that it is not needed to set the proximity relations among all the considered object classes, which would lead to a quadratic increment in the time spent in their definition, but just between the objects that are more commonly encountered together. Thus, extending the previous example, the person could avoid the definition of the relation between computer screens and trash bins, since they seldom appear close in an office.

4.2. Generation of training samples

Upon the semantic knowledge encoded in the ontology, the system generates samples in the form of synthetic scenes following four steps (notice that the stage presented here does not involve the human participation):

1. **Inclusion of objects in the scene**. The set of objects that appear in

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Concept	has_frequency OfOccurrence	P(appearing)	Sample
Floor	high	0.8	appearing
Wall	high	0.8	appearing
Table_top	veryHigh	0.9	appearing
Table_side	low	0.25	not_appe.
Chair_back	high	0.8	not_appe.
Chair_seat	medium	0.6	appearing
Computer _screen	high	0.8	appearing

is_nearTo	Frequency	P(near)	Sample
Floor	null	0	not_near
Wall	high	0.75	near
Chair_seat	high	0.75	near
Computer_s creen	veryHigh	0.9	near

Figure 4: Left, example of discrete probability distributions built according to the `has_frequencyOfOcurrece` relation of each concept. These distributions determine which objects are included into the synthetic scenario. Right, context creation for an object of the class `table_top` according to the objects included in the synthetic scenario.

the synthetic scene is selected according to the relations `has_frecuency-OfOcurrence` defined in the ontology. For that, we use a discrete probability distribution that establishes the likelihood of the presence of each object. For example, following the `Table_top` definition where `has_frecuencyOfOcurrence=high`, such a probability distribution can be defined by the person as $P(Table\_top_{appearing}) = 0.8$ and $P(Table\_top_{notAppearing}) = 0.2$. Samples from these distributions are drawn, yielding the set of objects included in the scene as illustrated in figure 4-left. In this example the objects included are: parts of the floor and a wall, a table top, a chair seat and a computer screen.

2. **Object characterization**. The geometrical features of the objects included in the synthetic scene in the previous step are reified according to their concepts' definitions in the ontology. To this end, a Gaussian distribution, $N(\mu, \sigma)$, is considered for each defined concept and for each defined geometric property, i.e. `has_area`, `has_centroidHeight`, `has_elongation` and `has_orientation`, where the mean μ is the value of that concept for that property in the ontology, and the standard deviation σ is a quantification of the respective annotated variability. For instance, for the `has_area` property of the `Table_top` concept, the person implicitly encoded a Gaussian distribution with $\mu = 1$ and a *high* standard deviation, e.g. $\sigma = 0.75$. Then, samples drawn from these distributions are used as features of the included objects (see figure 5-

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

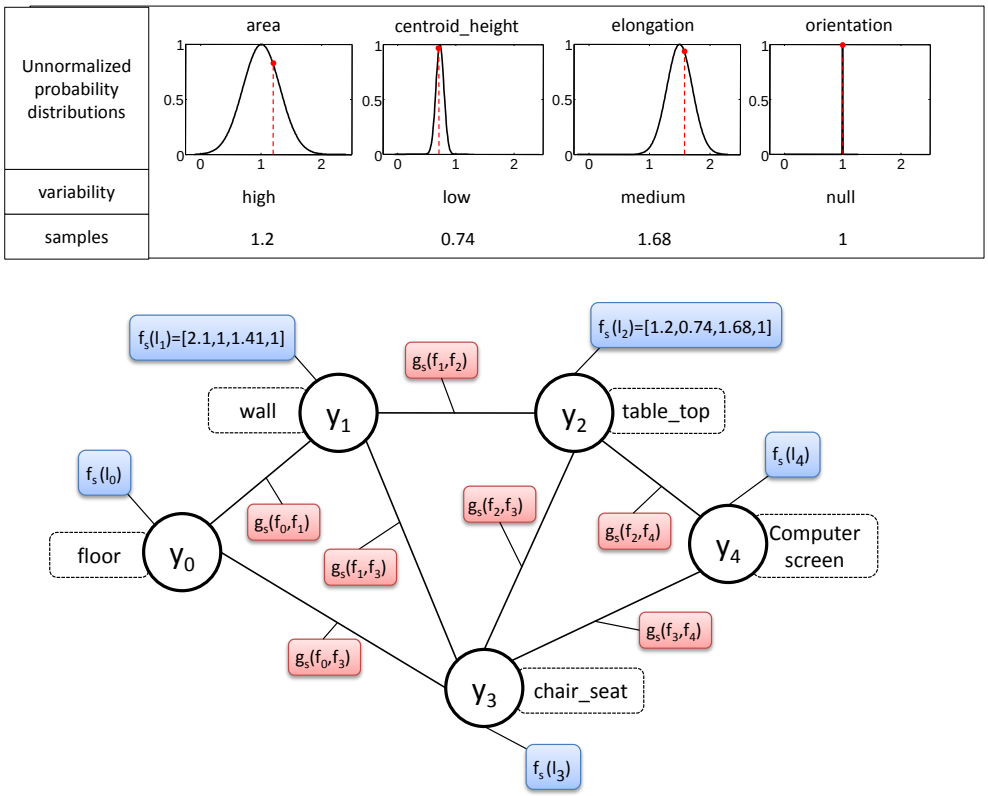

Figure 5: Top, samples drawn (red lines) from the probability distributions for an object of the class `Table_top`, built according to its geometrical restrictions and the annotated variability in the ontology (see figure 3-b). Bottom, graphical model that results from the objects included in figure 4-left and their generated relations.

top). These synthetic features are computed by the $f_s(l_i)$ function, where l_i is the class of the included object i. This function replaces $f(x_i)$ during the training phase (recall equation 1 in section 3).

3. **Context creation**. The contextual relations between the included objects are established according to the `is_nearTo` properties and their `frequency` annotations. For example, if the scene contains a `Table_top` and a `Chair_backRest`, they will be placed near one to another into the synthetic scene with a `high` probability, as stated by the ontology. Figure 4-right shows an example of the definition of the contextual relations for a `Table_top` object according to the objects previously included in the scene (see figure 4-left), its `is_nearTo` relations and

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

their `frequency` annotations.

4. **Context characterization**. Different features for the relations established in the previous step are computed, adding valuable contextual information. Examples of these features are: difference between centroid heights, perpendicularity, difference between areas, areas ratio, difference between elongations, etc. To compute them, the information produced in the *objects characterization* step is used. For example, if a `Table_top` with a height of 0.7 m. and a `Chair_backRest` showing a height of 0.32 m. are placed near in a synthetic scenario, their context can be characterized with the difference between the heights of their centroids: 0.38 m.
   Two additional binary features are considered to establish that an object is placed *on* or *under* other, according to the `is_on` and `is_under` relations of the ontology. Notice that these features characterize the context of a pair of objects that have been previously related in the synthetic scenario according to their proximity.
   The set of contextual features for objects (i, j) are yielded by the function $g_s(f_i, f_j)$, where $f_i = [f_s(l_i), l_i]$, being $f_s(l_i)$ the features computed in the *object characterization* step for object i, and l_i the class of that object. This function replaces the $g(x_i, x_j)$ one in equation 2 (section 3).

Figure 5-bottom shows the components of a synthetic scene produced by the steps described above in the form of a graphical model, compound of nodes representing the included objects, and edges stating their relations. Notice that the characterization of a `Table_top` illustrated in figure 5-top is in fact carried out by $f_s(l_2)$. As an example of context characterization, let's consider the context established by the objects `wall` (node y_1) and `table_top` (node y_2). Supposing that the contextual features employed are, for instance, difference between centroid heights, perpendicularity, *is on* and *is under*, then such a characterization is generated as $g_s(f_1, f_2) = [0.9, 1, 0, 0]$, which sets that: their centroids are separated by a vertical distance of 0.9 m.; given that the `wall` is vertical and the `table_top` is horizontal they are perpendicular; any object is located on or under the other one.

5. Evaluation

In order to evaluate our approach, we have trained a number of CRFs with synthetic data and assessed their suitability to recognize objects from: i)

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Figure 6: The mobile robot Rhodon gathering 3D data within an office room.

office scenarios within the UMA-offices dataset (section 5.1), and ii) office and home scenes within the NYU2 dataset (Silberman et al., 2012)(section 5.2).

5.1. Results with the UMA-offices dataset

The UMA-offices dataset was acquired with the mobile robot Rhodon, equipped with a Kinect device mounted on a pan-tilt unit (see figure 6), and entails 25 office environments from the University of Málaga. In the experiments, seven object classes were considered: $L = \{$*floor, wall, table, table_side, chair_back_rest, chair_seat* and *computer_screen*$\}$, and the ground-truth was provided by an human operator. It is worth to mention that the person that carried out the human elicitation process in the experiments (section 4.1) has worked in different office environments, but he did not visit the offices from the gathered dataset.

In our implementation, we rely on the UGM library (Schmidt, 2015) for training the CRF using the optimization of the pseudo-likelihood function (see section 3.1). Concretely, a Quasi-Newton method with Limited-Memory BFGS (Nocedal, 1980) is used, which is able to optimize complex objective functions with a high number of parameters.

The performance of CRFs trained with the proposed method is assessed

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

through the micro/macro precision/recall metrics (Anand et al., 2013) computed for the results yielded by the recognition process. Briefly, the *precision* of a given class of objects c_i is defined as the percentage of objects recognized as belonging to c_i that really belong to that class. Let $recognized(c_i)$ be the set of objects recognized as belonging to the class c_i, $gt(c_i)$ the set of objects of that class in the ground-truth, and $|\cdot|$ is the cardinality of a set, then the *precision* of the classifier for the class c_i is defined as:

$$precision(c_i) = \frac{|recognized(c_i) \bigcap gt(c_i)|}{|recognized(c_i)|} \tag{8}$$

On the other hand, the *recall* of a class c_i expresses the percentage of the objects that belonging to c_i are recognized as members of that class:

$$recall(c_i) = \frac{|recognized(c_i) \bigcap gt(c_i)|}{|gt(c_i)|}. \tag{9}$$

Precision and recall are metrics associated to a single class. It is also of interest to know the performance of the proposed method for all the considered classes. This can be measured by adding the so-called macro/micro concepts. *Macro precision/recall* represents the average value of the precision/recall for a number of classes, and it is defined in the following way:

$$macro\_precision = \frac{\sum_{i \in L} precision(c_i)}{|L|} \tag{10}$$

$$macro\_recall = \frac{\sum_{i \in L} recall(c_i)}{|L|} \tag{11}$$

Finally, *micro precision/recall* represents the percentage of objects in the dataset that are correctly recognized with independence of their belonging class, that is:

$$micro\_precision(c_i) = \frac{\sum_{i \in L} |recognized(c_i) \bigcap gt(c_i)|}{\sum_{i \in L} |recognized(c_i)|} \tag{12}$$

$$micro\_recall(c_i) = \frac{\sum_{i \in L} |recognized(c_i) \bigcap gt(c_i)|}{\sum_{i \in L} |gt(c_i)|} \tag{13}$$

Since we assume that objects belong to a unique class, then $\sum_{i \in L} |gt(c_i)| = \sum_{i \in L} |recognized(c_i)|$, and consequently the computation of both micro precision/ recall metrics gives the same value.

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Table 3: Results of the recognition process with different sets of pairwise features (configurations) and methods for the UMA-offices dataset. For the convenience of the reader, these features, previously listed in table 1, are: i_1–Perpendicularity, i_2–on/under relation, i_3–Vertical distance of centroids, i_4–Ratio between areas, and i_5–Ratio between elongations. The features employed in each configuration are: #1=$\{None\}$, #2=$\{i_1, i_2, i_3\}$, #3=$\{i_1, i_2, i_3, i_4\}$, and #4=$\{i_1, i_2, i_3, i_4, i_5\}$.

Method	Metric	Configurations #1	#2	#3	#4
CRF trained with synthetic data	micro p./r.	81.82	**90.91**	86.06	84.85
	macro p.	80.17	**89.25**	84.91	81.82
	macro r.	83.78	**89.99**	86.69	83.95
CRF trained with real data Koller and Friedman (2009)	micro p./r.	83.19	**87.50**	86.65	84.47
	macro p.	81.93	**85.84**	85.19	81.90
	macro r.	82.76	**86.36**	85.72	82.46

In our experiments we have trained five CRFs using the same synthetic dataset that comprises 1000 training samples including a total of 7170 objects and 16700 relations among them. CRFs differ in the combination of the selected pairwise features (configurations), aiming to analyze their suitability to the given environment.

The trained CRFs with synthetic data have been used to recognize the objects from the UMA-offices dataset. The results of the recognition process using the above metrics are shown in table 3. Observe that the achieved micro precision/recall is above 81%, with a best value of 90.91% for the configuration #2. Figure 8 shows some scene objects recognized with this configuration, while figure 7-left illustrates its confusion matrix. Note that in this case, the most challenging class to recognize is `table_side`, since it may not be clearly differentiated from other object classes like `chair_back`. Next, we highlight some meaningful comparisons and results of our approach

Comparison with state-of-the-art methods. We have compared the results of our method with two state-of-the-art alternatives: i) a standard formulation of a CRF trained and tested with real data (Koller and Friedman, 2009), and ii) the CRF presented in Xiong and Huber (2010). The results for both recognition systems were obtained through a 5-fold cross-validation and average process using the UMA-offices dataset. Such a process firstly splits the 25 offices into 5 groups. Then, four of these groups are used for training, and the remaining one for testing. This process is repeated five

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

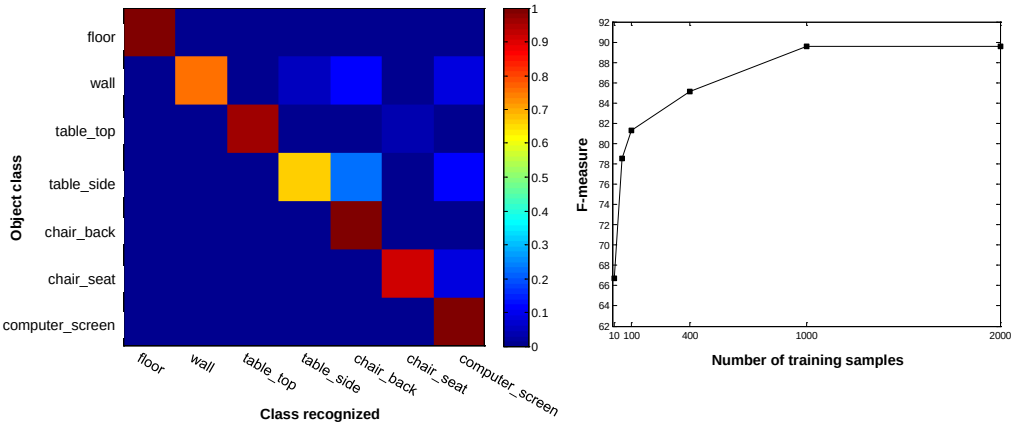

Figure 7: Left, confusion matrix that relates the ground truth to the recognition results in the second configuration. Right, influence of the number of training samples on the recognition success as it is measured by the F-measure.

times shifting the group used for testing, and finally the results are averaged. Table 3 shows the results for the evaluation with the CRFs in Koller and Friedman (2009), while the CRF with the configuration presented in Xiong and Huber (2010) achieved a micro p./r. of 82.46%. These figures reveal that CRFs trained with the proposed method can compete with, an even outperform the results of the other two state-of-the-art alternatives.

How much does the context relations contribute to the recognition performance? We have trained a CRF that does not consider pairwise factors, i.e., only taking into account the geometric properties of the planar patches (unary factors). The recognition results of using this CRF correspond to the first configuration in table 3, which shows a significantly lower success than the other configurations exploiting contextual relations.

What pairwise features are more discriminative? Notice that, in the results shown in table 3, the best ones are obtained when using perpendicularity, on/under and centroid height difference relations (configuration #2), whereas the inclusion of the area and elongation ratios (configurations #3 and #4) deteriorates the method performance. This indicates that both features have a low discriminant capability, influencing negatively to the recognition process. It is important to underscore that this conclusion only holds for systems employing the set of object classes L, so these contextual features could be useful in other applications or domains relying on a different set of object classes.

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

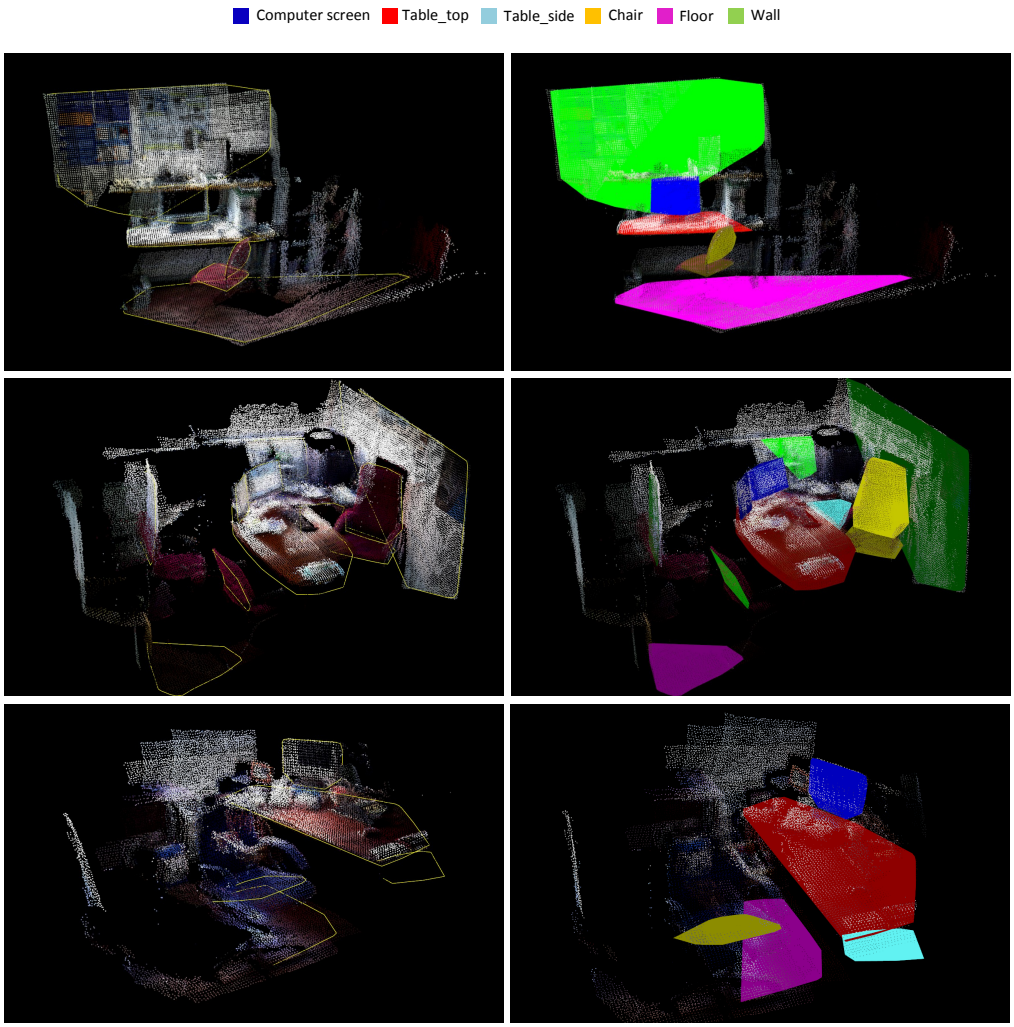

Figure 8: Examples of scene object recognitions performed by our method. Left column, observed scenes with the detected planar patches delimited by yellow lines. Right column, recognition results of such scenes.

How much does the size of the training dataset affect the recognition performance? Given that our method can generate an arbitrary number of samples, we have trained several CRFs with datasets of different sizes. To facilitate the comparison of their outcomes, the previous macro precision/recall metrics has been combined through the computation of their harmonic mean, also known as the $F-measure$. The harmonic mean, that

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

mitigates the impact of large measures and increments the influence of small values, is defined as follows:

$$F = 2 * \frac{macro\_precision * macro\_recall}{macro\_precision + macro\_recall} \tag{14}$$

Figure 7 shows the results of such outcomes, where the F value increases from the 66.68 obtained with 10 training samples up to 89.61 with 1000 samples. Notice that in this experiment the improvement reaches an upper limit for 1000 samples. This result remarks the importance of using large datasets to properly capture the variability of the domain as well as the convenience of techniques to reduce the burden of data gathering.

Do the generated synthetic data capture actual object properties and relations? In order to test the validity of the synthetic data generated for training CRFs, that is, how well the elicited ontology and the proposed method capture the real world, we have employed a CRF trained with our approach for recognizing objects from both real and synthetic datasets. Concretely, we have considered the CRF with configuration #2, the 25 offices from the UMA-offices dataset, and 25 synthetic scenarios generated with the approach described in section 4.2. The performance testing with the synthetic dataset yielded a micro precision/recall of 91.85%, a macro precision of 90.30%, and a macro recall of 90.39%. Note that these figures are similar to those obtained for the real dataset (see table 3, configuration #2), which reveals the suitability of both the ontology defined by the person and our approach for the generation of synthetic scenarios through the exploitation of semantic knowledge.

Computational performance. The training process, including the generation of synthetic samples, takes from 0.21 seconds when using 10 samples, up to 39.62 seconds for 1500 in a PC with an Intel®Core™$i5$ 3330 microprocessor at 3GHz and 8 GB DDR3 RAM memory at 1.6 GHz. Notice that the training process is performed only once, and does not take place during the robot operation. On the other hand, the inference process takes, on average, less than 0.2 milliseconds, which enables its integration in object recognition frameworks aiming to operate on-line.

Time saving using human elicitation plus synthetic samples generation. The results obtained in our experiments justify our claim that the proposed method can successfully replace the time-consuming and arduous tasks of gathering and processing real datasets. In order to also support its advantage for saving time/cost in the process, we have measured the time

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Table 4: Results of the recognition process with different sets of pairwise features (configurations) and methods for the NYU2 dataset. No pairwise features are used within configuration #1. #2 resort to i_1–Perpendicularity, i_2–on/under, relation, and i_3–Vertical distance of centroids.

Method	Metric	Configurations #1	#2
CRF trained with synthetic data	micro p./r.	76.23	**81.37**
	macro p.	73.72	**79.21**
	macro r.	76.32	**80.35**
CRF trained with real data Koller and Friedman (2009)	micro p./r.	74.21	**76.03**
	macro p.	65.57	**67.65**
	macro r.	66.70	**69.57**

consumed by the human elicitation and samples generation processes.

In our experiments, the human elicitation process for the office domain took 20 minutes, including the collection of the knowledge from the person and its codification into an ontology.

On the other hand, the time employed in the synthetic samples generation is negligible, since our method is capable of generating hundreds of samples in a less than a second (e.g., 1500 samples in $0.11sec.$). Thus, summing up the time spent for human elicitation, synthetic samples generation, and CRF training, our object recognition system can be ready to work in less than 21 minutes. Thereby, the presented methodology reduces dramatically the time required for training with real data, which involves the navigation of the robot through a number of locations (large enough to capture the variability of the domain), collecting the data, and its posterior processing. In our case, the gathering and processing of the 25 offices within the UMA-offices dataset took more than 7 hours, that is, 20 times higher than the time needed by our method.

5.2. Results with the NYU2 dataset

Our approach has been also evaluated considering 61 scenes from *office-environments*, and 200 *home-environment* scenes, all of them from the NYU2 dataset (Silberman et al., 2012).

Office-environments. For the tests within the office domain, two of the five CRFs trained during the evaluation with the UMA-offices have been

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

reused, concretely the ones with configurations #1 and #2. Notice that the same set of objects classes L has been considered.

Table 4 depicts the results of these tests. We can see how the integration of contextual information increments the micro p./r. value in a $\sim 5\%$. This is lower than the $\sim 9\%$ achieved with UMA-offices, which can be explained by the limited contextual information obtained from one-shot observations in NYU2 w.r.t. the multi-shot registered scenarios gathered in the UMA-offices dataset.

The performance of our approach has been also contrasted with: i) the results yielded by a standard CRF (Koller and Friedman, 2009) trained and tested with office data from NYU2, and ii) the CRF configuration from Xiong and Huber (2010), following again a 5-fold cross-validation and average methodology. The second row of table 4 shows the outcome of CRFs from Koller and Friedman (2009), while the configuration in Xiong and Huber (2010) reached a micro p./r. of 73.10% relying on unary features, and of 75.42% also integrating the pairwise ones. Both systems improve their results a $\sim 2\%$ when contextual information is introduced, however, they are still under the performance reached by the proposed methodology.

Home-environments. The aim of the testing with home scenes is to validate the applicability of the proposed approach to a different domain. For that, human elicitation has been used to define a new *home ontology*, publicly available at (http://goo.gl/mz51ho), which contains 20 object classes typically found in a home environment, e.g. bottle, cabinet, faucet, sink, toilet, sofa, pillow, bed, clothes, etc. These objects exhibit arbitrary shapes, so the recognition framework shown in figure 2 has been modified to work with object bounding boxes as geometric primitives, instead of the planar patches used in offices. In this case, the following properties replace those in table 2 for defining objects' concepts: `hasBiggestArea`, `hasColorVariation`, `hasElongation`, `hasHeight`, `hasOrientation`, `hasSize` and `isPlanar`. The contextual relations were codified in the same way as with the office ontology (recall section 4.1).

The resultant ontology was exploited to generate synthetic training data, and two CRF were tuned. The first CRF considers the following unary features to characterize an object: orientation, planarity, and size of its bounding box, area of its two principal directions, height from the floor, and color hue variation, and the second CRF also includes contextual relations characterized by: difference between principal directions, vertical distance of centroids, volume ratio, connectivity and object-object compatibility. These

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

configurations yielded a micro p./r. of 64% and a 69.44% respectively.

Additionally, a CRF following the standard formulation (Koller and Friedman, 2009) has been trained and tested through the above described 5-fold cross-validation and average process using the 200 home-environment scenes. In this case, the system achieved a 61.67% of micro p./r. relying only on unary features, and a 65.42% also considering contextual relations. A comparison with the CRF from Xiong and Huber (2010), as conducted in the previous sections, does not make sense here since it relies on planar patches. These figures support our claim that the proposed training approach can be applied to different environments compound of objects showing arbitrary shapes.

6. Conclusions and Future Work

Collecting real data for training object recognition systems is a highly time-consuming and cumbersome task, since the gathered data must be representative enough of the given domain. The approach presented in this paper overcomes this issue by replacing the data gathering task with the generation of synthetic samples. These samples implicitly capture the semantics of the scene by exploiting the knowledge codified in an ontology by a human. Our proposal has also the advantage of avoiding the processing of the collected sensorial information, which usually involves: segmentation, feature extraction, creation of contextual relations (if the recognition method leverages them), and finally regions' labeling by a human. In order to support our claim, we have trained and evaluated a number of Conditional Random Fields, with different sets of pairwise features and two datasets.

The results obtained in the conducted evaluations achieve a recognition success of $\sim 90\%$ within the UMA-offices dataset, and of $\sim 81\%$ and $\sim 69.5\%$ using office and home scenes from the NYU2 dataset respectively, revealing that the use of semantic knowledge can be exploited for the suitable training of recognition systems. Our approach has been also compared with other state-of-the-art approaches based on CRFs yielding a substantial improvement. A number of additional, related issues have been also addressed. Firstly, the discriminant capability of different sets of contextual features has been studied, showing their positive effect on the system performance. Also, the relation between the size of the training datasets and the system performance has been analyzed, obtaining the expected conclusions: the larger the dataset is, the better the system outcomes are. It has been also reckoned the

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

computational efficiency, evidencing the suitability of the proposed system for real time robot applications. Finally, we have studied the time saving gained with the use of human elicitation plus synthetic samples generation processes, resulting 20 times lower than the time spent in collecting real data from the UMA-offices dataset.

In the future we plan to exploit the symbolic representation of the recognized objects to perform higher-level robot tasks, such as efficient task planning or knowledge inference. We also plan to include temporal relations in the ontology as well as enabling crowdsourcing for the human elicitation process.

Acknowledgements

We are very grateful to our colleague E. Fernandez-Moral for providing us the implementation of the plane-based mapping algorithm, as well as for his support during the collection of the office dataset used to evaluate our method. This work has been funded by the Spanish grant program FPU-MICINN 2010 and the Spanish project "TAROTH: New developments toward a robot at home".

References

Anand, A., Koppula, H. S., Joachims, T., Saxena, A., Jan. 2013. Contextually guided semantic labeling and search for three-dimensional point clouds. In the International Journal of Robotics Research 32 (1), 19–34.

Baroni, M., Lenci, A., 2010. Distributional memory: A general framework for corpus-based semantics. Computational Linguistics 36 (4), 673–721.

Besag, J., 1986. On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society. Series B (Methodological) 48 (3), pp. 259–302.

Coradeschi, S., Saffiotti, A., 2003. An introduction to the anchoring problem. Robotics and Autonomous Systems 43 (2-3), 85–96.

Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A., 2005. Learning object categories from google's image search. In: IEEE International Conference on Computer Vision (ICCV 2005). Vol. 2. pp. 1816–1823 Vol. 2.

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Fernandez-Moral, E., Mayol-Cuevas, W., Arevalo, V., Gonzalez-Jimenez, J., 2013. Fast place recognition with plane-based maps. In: IEEE International Conference on Robotics and Automation (ICRA 2013). pp. 2719–2724.

Floros, G., Leibe, B., 2012. Joint 2d-3d temporally consistent semantic segmentation of street scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012). pp. 2823–2830.

Galindo, C., Fernandez-Madrigal, J., Gonzalez, J., Saffiotti, A., 2007. Using semantic information for improving efficiency of robot task planning. In: IEEE International Conference on Robotics and Automation (ICRA), Workshop on Semantic Information in Robotics. Rome, Italy.

Galindo, C., Fernandez-Madrigal, J., Gonzalez, J., Saffiotti, A., 2008. Robot task planning using semantic maps. Robotics and Autonomous Systems 56 (11), 955–966.

Galindo, C., Saffiotti, A., 2013. Inferring robot goals from violations of semantic knowledge. Robotics and Autonomous Systems 61 (10), 1131–1143.

Galleguillos, C., Belongie, S., Jun. 2010. Context based object categorization: A critical survey. Computer Vision and Image Understanding 114 (6), 712–722.

Günther, M., Wiemann, T., Albrecht, S., Hertzberg, J., 2013. Building semantic object maps from sparse and noisy 3d data. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013). pp. 2228–2233.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M., 2004. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission, World Wide Web Consortium.

Jancsary, J., Nowozin, S., Sharp, T., Rother, C., 2012. Regression tree fields - an efficient, non-parametric approach to image labeling problems. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2012). pp. 2376–2383.

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Kahler, O., Reid, I., Dec 2013. Efficient 3d scene labeling using fields of trees. In: IEEE International Conference on Computer Vision (ICCV 2013). pp. 3064–3071.

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

Kunze, L., Kumar, K., Hawes, N., 2014. Indirect object search based on qualitative spatial relations. In: IEEE International Conference on Robotics and Automation (ICRA 2014). Hong Kong, China.

Le, D.-T., Uijlings, J., Bernardi, R., 2013. Exploiting language models for visual recognition. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Seattle, Washington, USA, pp. 769–779.

Lowe, D. G., Nov. 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2), 91–110.

Mottaghi, R., Ranganathan, A., Yuille, A. L., 2011. A compositional approach to learning part-based models of objects. In: IEEE International Conference on Computer Vision Workshops (ICCV 2011 Workshops). pp. 561–568.

Nister, D., Stewenius, H., 2006. Scalable recognition with a vocabulary tree. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 2. pp. 2161–2168.

Nocedal, J., 1980. Updating quasi-newton matrices with limited storage. In: Mathematics of Computation. Vol. 35. pp. 2376–2383.

Nowozin, S., Rother, C., Bagon, S., Sharp, T., Yao, B., Kohli, P., 2011. Decision tree fields. In: IEEE International Conference on Computer Vision (ICCV 2011). pp. 1668–1675.

Nüchter, A., Hertzberg, J., 2008. Towards semantic maps for mobile robots. Robots and Autonomous Systems 56 (11), 915–926.

Quattoni, A., Collins, M., Darrell, T., 2004. Conditional random fields for object recognition. In: Advances in Neural Information Processing Systems. MIT Press, pp. 1097–1104.

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

Ren, X., Bo, L., Fox, D., 2012. Rgb-(d) scene labeling: Features and algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012). pp. 2759–2766.

Schmidt, M., 2015. UGM: Matlab Code for Undirected Graphical Models. `http://www.cs.ubc.ca/~schmidtm/Software/UGM.html`, [Online; accessed 28-April-2015].

Silberman, N., Hoiem, D., Kohli, P., Fergus, R., 2012. Indoor Segmentation and Support Inference from RGBD Images. In: Proc. of the 12th European Conference on Computer Vision (ECCV 2012). Springer-Verlag, Berlin, Heidelberg, pp. 746–760.

Speer, R., Havasi, C., 2013. Conceptnet 5: a large semantic network for relational knowledge. In: The Peoples Web Meets NLP. Theory and Applications of Natural Language. Springer, pp. 161–176.

Uschold, M., Gruninger, M., 1996. Ontologies: principles, methods and applications. The Knowledge Engineering Review 11, 93–136.

Valentin, J., Sengupta, S., Warrell, J., Shahrokni, A., Torr, P., 2013. Mesh based semantic modelling for indoor and outdoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013). pp. 2067–2074.

Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001). Vol. 1. pp. 511–518.

Xiang, Y., Zhou, X., Liu, Z., Chua, T.-S., Ngo, C.-W., 2010. Semantic context modeling with maximal margin conditional random fields for automatic image annotation. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3368–3375.

Xiong, X., Huber, D., 2010. Using context to create semantic 3d models of indoor environments. In: In Proceedings of the British Machine Vision Conference (BMVC 2010). pp. 45.1–11.

Zhou, K., Zillich, M., Zender, H., Vincze, M., 2012. Web mining driven object locality knowledge acquisition for efficient robot behavior. In: IEEE/RSJ

Accepted author version.
Final version published in Knowledge-Based Systems
DOI: https://doi.org/10.1016/j.knosys.2015.05.032

International Conference on Intelligent Robots and Systems (IROS 2012). pp. 3962–3969.