Robotics and Autonomous Systems 61 (2013) 932–947

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Indoor scene recognition by a mobile robot through adaptive object detection

P. Espinace [a,*], T. Kollar [b], N. Roy [b], A. Soto [a]

[a] *Department of Computer Science, Pontificia Universidad Catolica de Chile, Chile*

[b] *Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, United States*

HIGHLIGHTS

- We use common objects as an intermediate representation for indoor scene recognition.
- We frame our method as a generative probabilistic hierarchical model.
- We use rich sources of online data to populate the terms that compose our model.
- We use concepts from information theory to propose an adaptive scheme.
- Results show that the proposed approach outperforms other state-of-the-art techniques.

ARTICLE INFO

Article history:
Received 15 March 2012
Received in revised form
5 January 2013
Accepted 2 May 2013
Available online 18 May 2013

Keywords:
Scene recognition
Mobile robotics
Visual recognition
3D cues

ABSTRACT

Mobile robotics has achieved notable progress, however, to increase the complexity of the tasks that mobile robots can perform in natural environments, we need to provide them with a greater semantic understanding of their surrounding. In particular, identifying indoor scenes, such as an Office or a Kitchen, is a highly valuable perceptual ability for an indoor mobile robot, and in this paper we propose a new technique to achieve this goal. As a distinguishing feature, we use common objects, such as Doors or furniture, as a key intermediate representation to recognize indoor scenes. We frame our method as a generative probabilistic hierarchical model, where we use object category classifiers to associate low-level visual features to objects, and contextual relations to associate objects to scenes. The inherent semantic interpretation of common objects allows us to use rich sources of online data to populate the probabilistic terms of our model. In contrast to alternative computer vision based methods, we boost performance by exploiting the embedded and dynamic nature of a mobile robot. In particular, we increase detection accuracy and efficiency by using a 3D range sensor that allows us to implement a focus of attention mechanism based on geometric and structural information. Furthermore, we use concepts from information theory to propose an adaptive scheme that limits computational load by selectively guiding the search for informative objects. The operation of this scheme is facilitated by the dynamic nature of a mobile robot that is constantly changing its field of view. We test our approach using real data captured by a mobile robot navigating in Office and home environments. Our results indicate that the proposed approach outperforms several state-of-the-art techniques for scene recognition.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Mobile robotics is reaching a level of maturity that is starting to allow robots to move out of research labs [1,2]. Despite the advances, current robots still have very limited capabilities to understand their surroundings. For example, most robots still represent their environment using low-level maps usually limited to information about busy and empty spaces [3], low level visual landmarks [4,5], or specific structural constraints [6]. We believe that to increase the complexity of the tasks that mobile robots can perform in natural environments, there is an urgent need to provide them with a greater semantic understanding of their surroundings.

Vision appears as one of the most suitable sensor modalities to bridge the semantic gap of current mobile robots. The robustness and flexibility exhibited by most seeing beings is a clear proof of the advantages of a suitable visual system. Furthermore, recently the synergistic combination of computer vision and machine learning techniques [7–9] has opened a new promising paradigm to build robust seeing machines. As a relevant complement, we believe that the embodied and decision making nature of a mobile robot can facilitate the creation of robust perceptual systems. In effect, a mobile robot can actively explore the visual world looking for

[*] Corresponding author. Tel.: +56 223544440.
E-mail addresses: pespinac@ing.puc.cl (P. Espinace), tkollar@csail.mit.edu (T. Kollar), nickroy@csail.mit.edu (N. Roy), asoto@ing.puc.cl (A. Soto).

0921-8890/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.robot.2013.05.002

relevant views and features that can facilitate the recognition of relevant objects, scenes, or situations. These active visual behaviors are highly pervasive in the biological world, where from humans to bees perceptual actions drive visual inference [10,11].

In terms of semantic understanding of an environment, scene recognition appears as one of the most fundamental perceptual abilities, being a key contextual cue for scene understanding. Consequently, there is an extensive literature about scene recognition [12–14,4,15], mainly for outdoor environments. Traditionally, the main source of controversy has been between achieving scene recognition using low-level features to directly capture the gist of a scene [13] versus using higher level structures to capture intermediate semantic representations [14]. This intermediate representation can be obtained by using unsupervised learning methods [16,17] or more elaborated techniques, such as region segmentation [18,19]. In terms of robotics, as noticed before [12], identifying indoor scenes, such as an Office or a Kitchen, is a highly valuable perceptual ability that can facilitate the execution of high-level tasks by a mobile robot.

Following the motivations above, in this paper we propose a new technique for visual indoor scene recognition using a mobile robot. As distinguishing features, our approach is based on three main features. First, a probabilistic hierarchical representation that uses common indoor objects, such as Doors or furniture, as an intermediate semantic representation. Using this representation, we associate low-level visual features to objects by training object classifiers, and we associate objects to scenes by learning contextual relations among them. Second, we exploit the embedded nature of a mobile robot by using 3D information to implement a focus of attention mechanism. Using this mechanism, we can use 3D information to discard unlikely object locations and sizes. Third, we also exploit the embedded nature of a mobile robot and ideas from information theory to implement an adaptive strategy to search for relevant objects. Under this strategy, we use sequences of images captured during robot navigation to build a partial belief about the current scene that allow us to execute only the most informative object classifiers.

In terms of our hierarchical probabilistic approach, previous works for scene recognition do not perform well in the type of scenes usually visited by an indoor mobile robot. As we demonstrate in this paper, and has also been recently demonstrated by [15], most previous techniques for scene recognition show a significant drop in performance for the case of indoor scenes. This can be explained by the fact that, as opposed to outdoor scenes, indoor scenes usually lack distinctive local or global visual textural patterns. We believe that the use of objects as an explicit intermediate representation can help to improve this situation. Furthermore, in terms of a machine learning approach to scene recognition, the use of an intermediate representation based on common objects, with a clear semantic meaning, facilitates the acquisition of training data from public web sources, such as the Flickr website [20].

In terms of our focus of attention mechanism, previous works on scene recognition are based on a passive operation, where scene recognition is usually based on the independent analysis of single images. In contrast, in our case as our robot navigates through an environment, it can constantly provide new views of objects that enrich acquired information. In particular, we are able to dramatically reduce image processing time by incorporating structural and geometrical 3D information using a range sensor. This sensor helps us to filter spurious or false positive detections and to implement a focus of attention mechanism that can identify suitable scales and image areas to search for relevant objects. This strategy, in combination with an efficient feature extraction procedure such as integral channel features [21], brings overall robot operation closer to real time performance.

Finally, in terms of our adaptive method to search for relevant objects, there have been a few recent methods that also use objects in the scene recognition process [22–25], however, besides the fact that they are based on a different mathematical representation, they only apply a fix scheme to search for relevant objects in the scene. Clearly, a fix policy to search for a relevant object does not scale properly in terms of the number of potential objects. In our case, we use concepts from information theory to add to our method a planning strategy to search for likely objects. This allows us to adaptively search for objects according to our partial belief about the current scene. The key idea is to execute only the most informative object classifiers, based on the intuition that it is often enough to find a subset of the available objects to recognize a scene with high confidence. This helps our approach to scale efficiently in terms of the number of potential objects in indoor environments.

Accordingly, the main contributions of this work are: (i) A new hierarchical probabilistic model for indoor scene recognition based on the detection of relevant common objects, (ii) A new focus of attention mechanism based on a 3D range sensor that fully exploits the embedded nature of a mobile robot by directly measuring physical properties of objects such as size, height, and range disparity, (iii) A new adaptive methodology that allows us to execute at each time only the most informative object classifiers, and (iv) An empirical evaluation of the proposed method, showing significant advantages with respect to several alternative techniques. As an additional contribution, we facilitate further research based on visual and depth information by making available online the codes and datasets used in our experiments.

The rest of this paper is organized as follows. Section 2 discusses relevant previous work on visual scene recognition. Section 3 presents main details of our hierarchical probabilistic model for indoor scene recognition. Section 4 provides implementation details about main probability terms involved in our model. Section 5 presents our main results and a comparison with alternative approaches. Finally, Section 6 presents the main conclusions of this work and future avenues of research.

2. Related work

Scene recognition, also known as scene classification or scene categorization, has been extensively studied in areas such as cognitive psychology and computer vision [26,27]. In terms of cognitive psychology, previous studies have shown that humans are extremely efficient in capturing the overall gist of natural images, suggesting that intermediate representations are not needed [26]. Consequently, early methods for scene recognition are mostly based on holistic models. These approaches extract low-level features from the complete image, such as color or texture, and use those features to classify different scene categories. Vailaya et al. use this approach to classify city vs. landscape images [28]. Later, they extend the method using a hierarchical classification scheme, where images are first classified as indoor or outdoor scenes [29]. Also using low-level global features, Chang et al. estimate a belief or confidence function over the available scene labels [30]. During training, one classifier is built for each available scene category, then, all classifiers are applied to each test image, computing a confidence score with respect to each possible scene. Ulrich and Nourbakhsh use color histograms as the image signature and a k-nearest neighbors scheme for classification [12]. They apply their method to topological localization of an indoor mobile robot, but retraining is needed for each specific indoor environment. In this sense, an important disadvantage of holistic methods based on global image features is a poor generalization beyond training sets.

More robust holistic approaches use semantic representations or extract low-level global image signatures from selective parts of the input image. Oliva and Torralba use an image representation based on features such as naturalness or openness, that represent dimensions in a space that they call spatial envelope [13]. These features are computed using coarsely localized spectral information. Siagian and Itti build image signatures by using orientation, color, and image intensity visual saliency maps [4]. These maps are also shared by a focus of attention mechanism [31]. They test their approach by recognizing scenes using an outdoor mobile robot.

In terms of methods based on local image features, early approaches use a straightforward extension of global models, where the input image is broken into local blocks or patches. Features and classifiers are applied to each of the blocks and then combined through a voting strategy [32], or a mixture of probabilistic classifiers [33]. As in the case of global features, these local methods also suffer from poor generalization capabilities.

As an alternative, some works base the scene detection on the identification of local image regions such as sky, grass, or mountains [18,19]. To obtain the relevant semantic regions, these methods rely on image segmentation techniques. Individual classifiers are then applied to label each segmented region. Unfortunately, these approaches inherit the poor performance of segmentation algorithms, a still open problem for the computer vision community. The segmentation problem is particularly relevant for the case of indoor scenes, where the presence of a large number of objects usually produces scenes with significant clutter that are difficult to segment.

In general, the main problem with the methods described above has been their inability to generalize from training data to new scenes [27]. As discussed in [15], this problem has been particularly relevant for the case of indoor scenes. Additionally, in some cases, the use of elaborated manual strategies to identify relevant intermediate scene properties [13,34] limits the scalability of such techniques.

Recent approaches have achieved good results in scene classification by using intermediate representations and bag-of-words schemes. Fei-Fei and Perona recognize scenes using an intermediate representation that is provided by an adapted version of the Latent Dirichlet Allocation (LDA) model [14]. Bosch et al. [35] and Sivic et al. [36] achieve scene classification by combining probabilistic Latent Semantic Analysis (pLSA) with local invariant features. Lazebnik et al. modify bag-of-words representations by using a spatial pyramid that divides the image into increasingly fine sub-regions with the idea of capturing spatial relations among different image parts [37]. As we mentioned before, these techniques show a significant drop in performance for the case of indoor scenes [15].

Recently, there has been significant effort in improving scene recognition for the case of indoor scenes. The growing industry of service robotics, where high-level human–robot interaction is a key fact, has motivated several works that seek to obtain relevant high-level information about the contents of an image. Quattoni and Torralba propose an indoor scene recognition algorithm based on combining local and global information [15]. They test their approach using 67 indoor image categories with results that outperform current approaches for the case of indoor scenes. Interestingly, although they do not explicitly use objects in their approach, they remark that some indoor scenes might be better characterized by the objects they contain, indicating that object detection might be relevant to improve scene recognition for the case of indoor environments. Following this idea, recent work has incorporated objects as a key element to improve scene recognition in indoor environments [22–24]. Li et al. represent an image as a scale-invariant response map of a large number of pre-trained generic object detectors [25]. This representation is suitable for several visual tasks. Pronobis et al. build an approach for scene recognition with applications to mobile robot localization. This is based on the extraction of spatial semantic concepts from general place appearance and geometry. They use their approach to obtain relevant high-level information for mobile robot navigation, such as a semantic map [38,39]. In contrast to our approach, these previous methods do not benefit from the embedded nature of a mobile robot, for example by using 3D structural information or a sequential adaptive object detection scheme.

In terms of object recognition, Viola and Jones show that it is possible to achieve real time category-level object recognition without relying on image segmentation. Instead, they use a sliding-window approach in conjunction with a focus of attention mechanism [8]. Dalal and Triggs introduce a histogram of oriented gradient (HOG) features to represent object categories. Also they use a sliding-window approach to detect object instances [40]. Felzenszwalb et al. develop an approach based on mixtures of multiscale deformable part models that uses a new method for discriminative training with partially labeled data, achieving outstanding results [9]. Recently, they enhance their approach by using cascade object detection [41]. Furthermore, Helmer and Lowe show the benefits of using 3D information to improve object recognition [42].

In terms of robotics, besides the fact that some of the methods described above are applied to this field, extensive work has been done in the case of topological localization using visual landmarks [5,43]. The main limitation of these approaches is that landmarks are usually environment specific, thus, generalization to different places produces poor results. In terms of object recognition, there has been work related to detecting relevant structures and objects in outdoor urban scenes [44,45].

Finally, it is worth mentioning that Bosch et al. [27] provide a full bibliographic review about the topic of scene recognition (up to 2007), including a deeper description of some of the methods mentioned above.

3. Problem formulation

In this section we present the mathematical formulation behind our method to use objects as an intermediate semantic representation between low-level features and high level scene concepts. First, we present the core of our method considering only visual features and leaving aside 3D properties. Then, we show how 3D geometrical properties can be incorporated to enhance our formulation. Afterwards, we provide a mathematical approximation that makes our method computationally feasible. Finally, we use information theory to build an adaptive scheme to guide the search for informative objects.

3.1. Scene recognition using visual features

In order to model our scene recognition approach, we include the following terms:

- Let ξ be a scene type, $\xi \in \varXi$.
- Let $s \in \{1, \dots, S\}$ be an object class.
- Let $o_s \in [0, 1]$ indicate the presence/absence of instances of objects of class s in a given scene.
- Let $p(\xi|o_s)$ be the probability that ξ is the underlying scene, given that an object of class o_s is present in the scene.
- Let I be an image.
- Let $w_i, i \in \{1, \dots, L\}$ be a rectangular window that covers a specific part of image I and defines an object location.
- Let $c_{w_i} \in \{0, \dots, S\}$ be the output of an object classifier c when applied to image location w_i. Output 0 indicates that no object is found.
- Let $c_{1:w_L}$ be a vector describing the outputs of L classifiers calculated over a set of L windows.
- Let $f_{w_i}^{j}$ be visual feature j extracted from image window w_i.
- Let $\vec{f}_{w_i}$ be a vector describing the complete set of visual features extracted from image window w_i.

- Let $\vec{f}_{1:w_L}$ be the complete set of visual features calculated over L windows.

Given these terms, the probability of a place ξ given a set of features $\vec{f}_{1:w_L}$ is:

$$\begin{aligned} p(\xi|\vec{f}_{1:w_L}) &= \sum_{o_{1:S}} \sum_{c_{1:w_L}} p(\xi|o_{1:S}, c_{1:w_L}, \vec{f}_{1:w_L}) p(o_{1:S}, c_{1:w_L}|\vec{f}_{1:w_L}) \\ &= \sum_{o_{1:S}} \sum_{c_{1:w_L}} p(\xi|o_{1:S}) p(o_{1:S}|c_{1:w_L}) p(c_{1:w_L}|\vec{f}_{1:w_L}). \end{aligned} \tag{1}$$

Let us now consider $p(o_{1:S}|c_{1:w_L})$ in Eq. (1). Using a Naive Bayes approximation such that objects are independent given the classifier outputs, we have:

$$p(o_{1:S}|c_{1:w_L}) = \prod_s p(o_s|c_{1:w_L}). \tag{2}$$

Also, let us assume that we have detector models relating the presence of an object of class s to the output of a classifier c in any possible window, such that:

$$p(o_s = 1|c_{w_{(\cdot)}} = o_k) = p_{o_s, c_{o_k}} = 1 - p_{\bar{o}_s, c_{o_k}}. \tag{3}$$

Then, considering that $p(o_s|c_{1:w_L}) = p(o_{s,w_1} \cup \cdots \cup o_{s,w_L}|c_{1:w_L})$ and assuming that windows are independent, we have:

$$\begin{aligned} p(o_{1:S}|c_{1:w_L}) &= \prod_s \left[1 - \prod_k (p_{\bar{o}_s, c_{o_k}})^{n_k} \right]^{o_s} \\ &\quad \times \left[\prod_k (p_{\bar{o}_s, c_{o_k}})^{n_k} \right]^{1-o_s}, \end{aligned} \tag{4}$$

where $k \in \{0, \ldots, S\}$ ranges over the possible classifier outputs and n_k is the number of classifications in $c_{1:w_L}$ with an output value o_k. $k = 0$ represents the case that no object is present in the respective image window. The assumption of independent windows is very strong and leads to overconfident posteriors, however, in practice we have not observed significant failures due to this approximation.

As an alternative to Eq. (4), when particular error models are not available for each possible classifier output, one can establish general error terms, such as:

$$\begin{aligned} &p(o_s = 1|c_{(\cdot)} = o_s) = p_{o_s, c_{o_s}} \\ &p(o_s = 1|c_{(\cdot)} \neq o_s) = p_{o_s, c_{\bar{o}_s}}. \end{aligned} \tag{5}$$

In this case, Eq. (4) is given by:

$$\begin{aligned} p(o_{1:S}|c_{1:w_L}) &= \prod_s [1 - (p_{\bar{o}_s, c_{o_s}})^{n_s} (p_{\bar{o}_s, c_{\bar{o}_s}})^{(L-n_s)}]^{o_s} \\ &\quad \ldots [(p_{\bar{o}_s, c_{o_s}})^{n_s} (p_{\bar{o}_s, c_{\bar{o}_s}})^{(L-n_s)}]^{1-o_s}. \end{aligned} \tag{6}$$

Let us now consider $p(c_{1:w_L}|\vec{f}_{1:w_L})$ in Eq. (1), assuming independence among the visual information provided by each window, we have:

$$p(c_{1:w_L}|\vec{f}_{1:w_L}) = \prod_i p(c_{w_i}|\vec{f}_{w_i}). \tag{7}$$

Therefore, using Eq. (4), we can finally express Eq. (1) as:

$$\begin{aligned} p(\xi|\vec{f}_{1:w_L}) &= \sum_{o_{1:S}} \sum_{c_{1:w_L}} p(\xi|o_{1:S}) \prod_s \left[1 - \prod_k (p_{\bar{o}_s, c_{o_k}})^{n_k} \right]^{o_s} \\ &\quad \ldots \left[\prod_k (p_{\bar{o}_s, c_{o_k}})^{n_k} \right]^{1-o_s} \prod_i p(c_{w_i}|\vec{f}_{w_i}). \end{aligned} \tag{8}$$

Note that this formulation can operate with any object detector able to classify objects from low-level visual features.

3.2. Adding 3D geometric information

In order to include 3D geometric information, we add the following terms to our model:

- Let D be a set of routines that calculate 3D geometric properties of an image.
- Let $d^j_{w_i}$ be the output of property j on window w_i.
- Let $\vec{d}_{w_i}$ be a vector describing the outputs of all the 3D geometric properties calculated over w_i.
- Let $\vec{d}_{1:w_L}$ be the complete set of geometric properties calculated over a set of L windows.

Given this information, our original problem in Eq. (1) becomes

$$\begin{aligned} p(\xi|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}) &= \sum_{o_{1:S}} \sum_{c_{1:w_L}} p(\xi|o_{1:S}, c_{1:w_L}, \vec{f}_{1:w_L}, \vec{d}_{1:w_L}) \\ &\quad \ldots p(o_{1:S}, c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}) \\ &= \sum_{o_{1:S}} \sum_{c_{1:w_L}} p(\xi|o_{1:S}) \\ &\quad \times p(o_{1:S}|c_{1:w_L}) p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}). \end{aligned} \tag{9}$$

In this case, $p(\xi|o_{1:S})$ and $p(o_{1:S}|c_{1:w_L})$ are as before. In terms of $p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L})$, we have:

$$p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}) = \prod_i p(c_{w_i}|\vec{f}_{w_i}, \vec{d}_{w_i}). \tag{10}$$

Applying Bayes rule and a conditional independence assumption, we can transform Eq. (10) into

$$p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}) \propto \prod_i p(\vec{d}_{w_i}|c_{w_i}) p(c_{w_i}|\vec{f}_{w_i}). \tag{11}$$

In our case, we use depth information to calculate three geometric properties: object size, object height, and object depth dispersion. We respectively denote these properties as: ds_{w_i}, dh_{w_i}, and dd_{w_i}. Then, $\vec{d}_{w_i} = \{ds_{w_i}, dh_{w_i}, dd_{w_i}\}$, so Eq. (11) becomes:

$$p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}) \propto \prod_i p(ds_{w_i}, dh_{w_i}, dd_{w_i}|c_{w_i}) p(c_{w_i}|\vec{f}_{w_i}). \tag{12}$$

Assuming conditional independence among the different geometric properties,

$$\begin{aligned} p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}) &\propto \prod_i p(ds_{w_i}|c_{w_i}) p(dh_{w_i}|c_{w_i}) \\ &\quad \ldots p(dd_{w_i}|c_{w_i}) p(c_{w_i}|\vec{f}_{w_i}). \end{aligned} \tag{13}$$

Finally, Eq. (8) becomes

$$\begin{aligned} p(\xi|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}) &= \alpha \sum_{o_{1:S}} \sum_{c_{1:w_L}} p(\xi|o_{1:S}) \\ &\times \prod_s \left[1 - \cdots \prod_k (p_{\bar{o}_s, c_{o_k}})^{n_k} \right]^{o_s} \left[\prod_k (p_{\bar{o}_s, c_{o_k}})^{n_k} \right]^{1-o_s} \\ &\times \prod_i \alpha p(ds_{w_i}|c_{w_i}) \ldots p(dh_{w_i}|c_{w_i}) \ldots p(dd_{w_i}|c_{w_i}) p(c_{w_i}|\vec{f}_{w_i}) \end{aligned} \tag{14}$$

where α is a constant that does not depend on the scene class. Also, the geometric properties are independent from visual information, thus, they can be used in combination with any chosen object classifier to enhance detection performance.

3.3. Reducing dimensionality

We can see that our mathematical formulation depends on two nested summations over combinations of objects and windows. In computational terms, we can estimate the complexity of our method as follows:

- The inner summation considers the presence of all possible objects in all possible windows, thus, its complexity is $O({N_{\text{obj}}}^{N_{\text{win}}})$, where N_{obj} is the number of objects being used, and N_{win} is the number of windows.
- The outer summation considers the presence of all possible objects in the scene, thus, its complexity is $2^{N_{\text{obj}}}$.
- Considering both summations, the complexity of the method is $2^{N_{\text{obj}}} \times {N_{\text{obj}}}^{N_{\text{win}}}$.

A complexity of $2^{N_{\text{obj}}} \times {N_{\text{obj}}}^{N_{\text{win}}}$ is intractable, particularly when N_{obj} may grow to the order of tens or hundreds and N_{win} is in the order of thousands. Fortunately, many of the cases considered in these summations are highly unlikely. For example, some of the cases may include non-realistic object combinations, or may consider objects that according to the classifiers are not present in the current image. Furthermore, we can use 3D information to discard unlikely object locations and sizes. Considering this, we can effectively reduce the computational complexity by discarding highly unlikely cases. To achieve this goal, we use a Monte Carlo technique to approximate the relevant summations in Eq. (14) using a sampling scheme based on a focus-of-attention mechanism. In this way, we focus processing only on likely hypothesis for each of the summations. In practice, as we will describe in our results, we observe that this approximation does not degrade the performance of the inference procedure.

For the outer summation we have

$$p(\xi|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}) = \sum_{o_{1:S}} \sum_{c_{1:L}} p(\xi|o_{1:S}) p(o_{1:S}|c_{1:w_L}) \ldots p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}). \tag{15}$$

We can take the first term out of the inner summation and using Bayes rule we obtain:

$$p(\xi|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}) = \sum_{o_{1:S}} \frac{p(o_{1:S}|\xi)p(\xi)}{p(o_{1:S})} \sum_{c_{1:w_L}} p(o_{1:S}|c_{1:w_L}) \ldots p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}). \tag{16}$$

This is equivalent to:

$$\sum_{o_{1:S}} p(o_{1:S}|\xi) F(o_{1:S}), \tag{17}$$

where

$$F(o_{1:S}) = \frac{p(\xi)}{p(o_{1:S})} \sum_{c_{1:w_L}} p(o_{1:S}|c_{1:w_L}) p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}). \tag{18}$$

We solve the summation by sampling from $p(o_{1:S}|\xi)$ and evaluating the samples in $F(o_{1:S})$. In the evaluation, we need to solve the inner summation.

For the inner summation we have

$$\sum_{c_{1:w_L}} p(o_{1:S}|c_{1:w_L}) p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}). \tag{19}$$

Again, we approximate the summation using a Monte Carlo scheme by sampling from $p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L})$ and evaluating the samples in $p(o_{1:S}|c_{1:w_L})$. Here, we use the combination $o_{1:S}$ that comes from the current sample of the outer summation. In order to sample from $p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L})$, we use our assumption of independence among windows:

- A combination $x \in c_{1:w_L}$ can be seen as a binary array of length L, where each element in the array represents the object that is present in one particular window (zero if nothing is present).
- A sample x_k can be obtained by getting a sample for each of the windows, $x_k = \{x_k^1, x_k^2, \ldots, x_k^L\}$, where each element x_k^i is obtained according to the probability distribution of the presence of objects in the corresponding window.
- For each window w_i, we build a multi-class probability distribution for the presence of objects in the window by joining a set of two-class object classifiers and normalizing afterwards.

3.4. Adaptive object search

So far, the presented formulation determines the scene type using S objects, thus, the multi-class window classifier c_{w_i} would have to run S binary object classifiers and normalize the outputs. Clearly, this approach does not scale properly with the number of object classes, as we need to execute a different classifier for each potential object category. Next, we show that we can estimate the current scene type with high confidence by running only a subset of the available classifiers. In particular, we use concepts from information theory to propose an adaptive scheme to guide the search for informative objects. Under this scheme, we use the current scene estimate to adaptively decide which object classifier to run next in order to maximize the reduction of current ambiguities. Formally, we slightly modify Eq. (1) by considering the case of running only a subset of n object classifiers ($n \leq S$):

$$p(\xi|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}, c^1_{1:w_L} \ldots c^n_{1:w_L}) = \sum_{o_{1:n}} \sum_{c^{1:n}_{1:w_L}} p(\xi|o_{1:n}) p(o_{1:n}|c^{1:n}_{1:w_L}) p(c^{1:n}_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}), \tag{20}$$

where $c^{1:n}_{1:w_L}$ represents the set of n classifiers associated to the detection of n different objects $o^1 \ldots o^n$ in L image windows w_i. Therefore, the formulation in Eq. (1) is equivalent to the case where $n = S$. To simplify our notation, from now on we refer to the classifiers as c^i dropping the subindex w_i.

Let us assume that we have an estimate of a scene probability distribution using $n - 1$ objects: $p(\xi|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}, c^1 \ldots c^{n-1})$. We would like to improve this estimate by using the most useful extra binary object classifier c^n. To do this, we could search for the classifier c^i that, when used, provides information that maximizes the information gain with respect to the current scene estimate. This is given by:

$$I[c^i] = [H(p(\xi|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}, c^1 \ldots c^{n-1})) - H(p(\xi|\vec{f}_{1:w_L}, \vec{d}_{1:w_L}, c^1 \ldots c^{n-1}, c^i))] \tag{21}$$

$$c^n = \operatorname*{argmax}_{c^i} I[c_i], \tag{22}$$

where $I[c^i]$ denotes the information gain provided by classifier c^i, which corresponds to the change of entropy H of the scene probability distribution given that we run classifier c^i. Unfortunately, computing this maximization involves running each possible object classifier, in which case we would rather prefer to run the full estimation using all the classifiers. To avoid this problem, we maximize the expected information gain of running each extra classifier. This is given by:

$$E\{I[c^i]\} = \sum_{c^i} p(c^i) I[c^i] \tag{23}$$

$$c^n = \operatorname*{argmax}_{c^i} E\{I[c^i]\}. \tag{24}$$

Here each binary classifier c^i can output either 0 or 1, and there is a confidence for that detection encapsulated in the term $p(c^i)$. As a consequence, we can calculate the expected values in Eq. (23) by estimating $p(c^i)$ without running the classifier by:

$$p(c^i) = \sum_{\xi \in \Xi} \sum_{o_i \in (0,1)} p(c^i|\xi, o_i)p(\xi, o_i)$$
$$= \sum_{\xi \in \Xi} \sum_{o_i \in (0,1)} p(c^i|\xi, o_i)p(o_i|\xi)p(\xi). \quad (25)$$

Next section presents details about how these terms are estimated.

4. Building the scene detector

In this section, we show how we compute each of the terms in the previous probabilistic model.

4.1. Category-level object detection

In this sub-section, we present our approach to category-level object detection and show how we compute $p(c_{1:w_L}|\vec{f}_{1:w_L}, \vec{d}_{1:w_L})$ in Eq. (11). As shown before, this term can be expressed as $\alpha p(\vec{d}_{w_i}|c_{w_i})p(c_{w_i}|\vec{f}_{w_i})$, therefore, we focus on these two sub-terms.

4.1.1. Computing $p(c_{w_i}|\vec{f}_{w_i})$

First, we apply an offline training procedure to obtain classifiers for each object category. We collect a representative dataset using selected images from 3 main sources: Label Me [46], Caltech 101 [47], and Google Images. Then, we extract a group of features for each training instance. We explore a large set of potentially relevant features to increase the hypothesis space, and rely on learning to prune the search for relevant features. Specifically, we use a pyramidal decomposition, similar to the approach in [48]. This decomposition computes the same bank of features at different image patches within a single image. This allows us to extract global and local information from each object instance. In our approach we use a 3-level pyramid, obtaining a total of 21 image patches per object instance.

In order to extract features that can be efficiently computed, we use histograms based on Integral Channel Features [21]. These are calculated using integral images from several image channels built from linear and non-linear transformations of the input image. Each of the resulting histograms is considered as one feature for our method.

In our implementation, we first compute 2 image maps: grayscale and fine-grained saliency [49]. We call these the base maps. Afterwards, we use the base maps to apply several transformations obtaining the following image channels:

- Base channels: these correspond directly to the initial base maps (2 channels total).
- Gabor channels: we apply Gabor filters to the base maps, given by 2-D Gaussian-shaped bandpass filters with dyadic treatment of the radial spatial frequency range and multiple orientations. We use 4 different orientations for each base map (8 channels total).
- Gradients channels: implemented in a similar way to HOG features but for each base channel we separate magnitude and orientation information (4 channels total).
- LBPs channels: a measure of texture that uses local appearance descriptors. We use 2 different radial distances for each base image (4 channels total).

The above processing produces a total of 18 integral histograms, one for each computed channel. The procedure to obtain the integral histograms consists of building one integral image for each possible gray value (256 for one byte depth channels), that summarizes the information of the corresponding gray value in the channel. For base maps, Gabor filters, and gradients, we build 6 different histograms, that differ in the number of bins, obtaining a total of 12 histograms for plane images, 48 histograms for Gabor filters, and 24 histograms for gradients. In the case of LBPs, we use the standard 256 bins LBP histogram and a uniform LBP histogram for each image, obtaining 8 LBP histograms, for an overall total of 92 histograms for each of the 21 image patches.

The integral histograms allow us to quickly compute an image histogram for any of the 21 image patches and for any required number of bins, thus, we can build features based on fine histograms (many bins) and coarse histograms (few bins). Our experiments show that both coarse and fine histograms can be useful under different situations. As we explain later, a feature selection procedure chooses the best histograms for classification.

With the previous feature extraction procedure, we obtain a total of 1932 features for each object instance. It is important to note that after building an integral histogram, the computational cost of computing an associated image histogram for any image patch and fixed number of bins is $O(1)$, as it only requires arithmetic operations over the integral histogram values. Therefore, if the features selected for a single object classifier or for several different object classifiers share the same base integral histogram, those features can be quickly computed in constant computational time after the initial integral histogram computation. This is very important for our implementation as it provides an important speed-up when running a group of object classifiers.

Using the available features, we train category-level object classifiers using the AdaBoost algorithm [50]. For each object classifier, our implementation builds one weak classifier for each computed histogram, each of them based on a random forest [51] that considers the histogram bins as inputs for classification. In this sense, we use the building feature selection properties of AdaBoost to select just a reduced subset of the potential 1932 weak classifiers. At execution time, we apply the classifiers using a sliding window procedure that allows us to compute $p(c_{w_i}|\vec{f}_{w_i})$, where we use the normalized output of the AdaBoost classifier as an estimate of this probability.

4.1.2. Computing $p(\vec{d}_{w_i}|c_{w_i})$

To obtain this term we use a 3D Swiss Ranger that provides a pixel level estimate of the distance from the camera to the objects in the environment (depth map). Given an image and its corresponding depth map, we use the camera parameters and standard projective geometry to calculate features $\vec{d} = \{ds, dh, dd\}$ for each candidate window containing a potential object, where *ds* refers to object size given by width and height, *dh* is the object altitude given by its distance from the floor plane, and *dd* is the object internal disparity given by the standard deviation of the depth values corresponding to the pixels inside the candidate window. Each of these individual properties has its associated term in our equations and their probabilities take the form of a Gaussian distribution with mean and covariance that are learned from training data,

$$ds_i|c_{w_i} \sim N(\mu_{ds}, \Sigma_{ds})$$
$$dh_i|c_{w_i} \sim N(\mu_{dh}, \sigma^2_{dh})$$
$$dd_i|c_{w_i} \sim N(\mu_{dd}, \sigma^2_{dd}).$$

Note that *ds* includes the height and width of the detection window, therefore is estimated using a 2-D Gaussian.

In order to take full advantage of 3D information, we use the geometric properties described before as a focus of attention mechanism. As seen in Eq. (12), the probability of the presence of an object in a window is a multiplication of a term that depends on 3D geometric features and a term that depends on visual features. We take advantage of this fact by using geometric properties as an initial classification step, quickly discarding image windows that contain inconsistent 3D information, such as a Door floating in the air. In our experiments, we find that by using geometric properties as an initial filtering step, we are able to reduce processing time by an average of 51.9% with respect to the case using just visual attributes.

4.2. Classifiers confidence

Given that an object has been detected at a specific window, we require an estimate of the confidence of that detection. These confidence values correspond to the term $p(o_{1:S}|c_{1:w_L})$ in Eq. (1). We estimate this term by counting the number of true-positives and false-positives provided by our classifiers on test datasets.

4.3. Prior of objects present in a scene

It is well known that some object configurations are more likely to appear in certain scene types than in others. As we show in [20], contextual prior information can be inferred from huge datasets, such as the Flickr website. In our method, we follow this approach by using representative images from this dataset, computing the frequency of each object configuration in these images according to their tags, and normalizing to obtain the probability distributions included in the term $p(\xi|o_{1:S})$ of our model. See [20] for more details.

4.4. Adaptive objects search terms

In order to implement our adaptive objects search strategy we need to estimate each of the terms on the right hand side of Eq. (25). The idea is to obtain an estimate for each term using an offline procedure, in order to avoid running all classifiers at execution time. Next, we provide details about how each of these terms are obtained.

To estimate the term $p(c^i|\xi, o_i)$, we need to obtain test data to evaluate each object classifier performance in images of each of the scene types. We use Flickr to obtain this test data, by collecting one test set for each object–scene couple using a group of images with the corresponding scene label that contain the object, and another group of images with the same scene label that do not contain the object. As an example, for the couple Lamp–Bedroom, we collect a large group of images that contain the Bedroom label and split it in a group that contains Lamp label and a group that does not contain Lamp label. Then, we execute the classifiers associated to each of the objects in these sets and evaluate their performance. This offline process provides a set of probability estimates $p'(c^i|\xi, o_i)$ for every classifier, object, and scene combination.

To estimate the term $p(o_i|\xi)$, we use a frequentist approach that counts the occurrence of each object in a group of images of each of the scene types. Once again, we use data from Flickr to obtain this term, by using one set of images for each scene type that contain the corresponding scene label and analyzing the frequency of appearance of each of the objects in these sets. As an example, for the same couple Lamp–Bedroom mentioned before, we use the same group of images that contain the Bedroom label and count the number of those images that contain the Lamp label. Thus, $p(o_i|\xi)$ for this couple would be the obtained count number divided by the total number of images with the Bedroom label.

Finally, to estimate the term $p(\xi)$, we need to obtain a prior of each scene probability. Here, we use the current scene estimate obtained by running previous $n-1$ object classifiers as these priors.

Results shown in the next section show that the approximation of $p(c^i)$ obtained by building the previous terms provides an efficient way to avoid running every object classifier. Furthermore, this reduction in processing time does not make an impact on scene recognition performance.

5. Results

In this section we present the results of our method performing several tests in different environments.

We first test our method using two different indoor Office environments: (i) Computer Science Department at Pontificia Universidad Catolica de Chile (DCC-PUC), and (ii) Computer Science and Artificial Intelligence Lab at Massachusetts Institute of Technology (CSAIL-MIT). In both environments, we select 4 different scenes or places where the method should compute a probability distribution given an input image: Office, Hall, Conference Room, and Bathroom. We use 7 different types of objects to estimate place probabilities: PC-Monitor, Door, Railing, Clock, Screen, Soap dispenser, and Urinal. Clearly, the objects in this set are more or less related to different places. These relationships are reflected in the corresponding priors that we estimate using training data. We train each object classifier using a set of 120 images manually selected from Google Images. To estimate the detection rates of each classifier, we use independent sets of 200 images, also manually selected from Google Images. We divide the experiments into two groups: (i) Single image tests, where we run our method using information from single images, and (ii) Sequence of images tests, where we run our method using sequences of images captured by a mobile robot while it navigates through the environment.

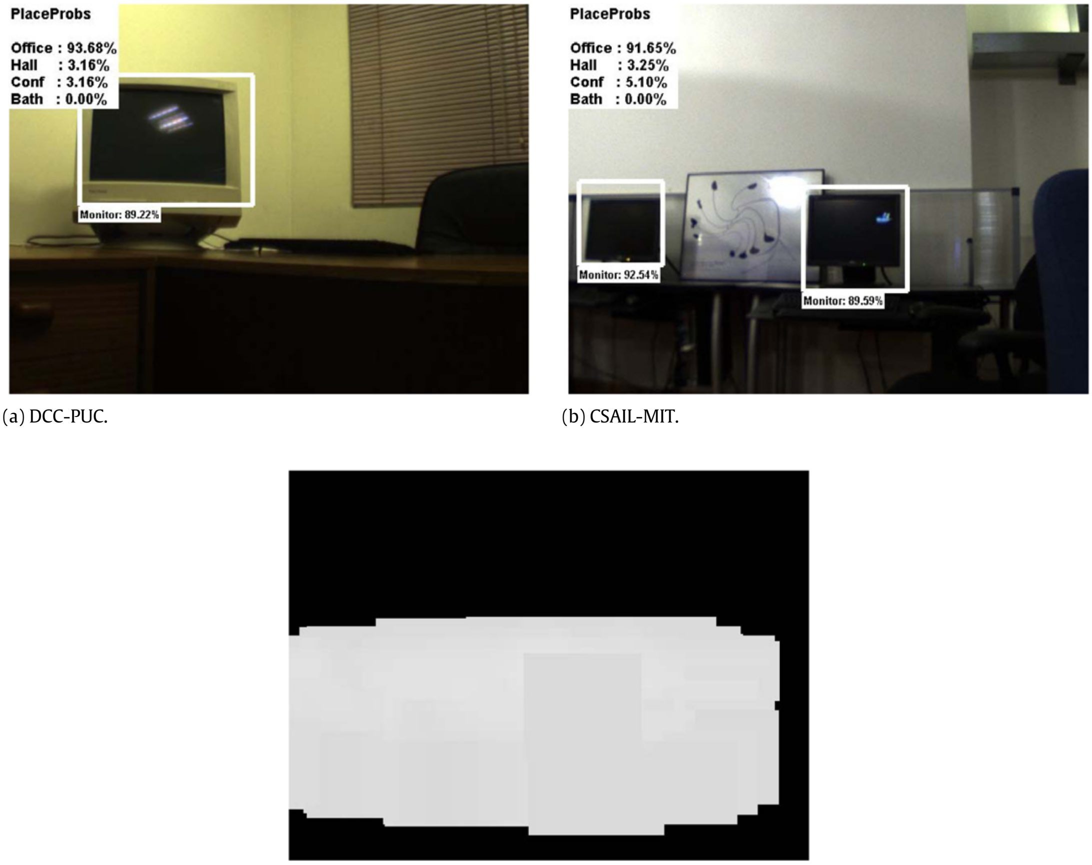
To further study the performance of our method, we also test the approach in an indoor home environment using a larger list of object classifiers that includes 12 different types of object categories: Bed, Chair, Sofa, Door, Dining Table, Lamp, Potted Plant, TV Monitor, Fridge, Microwave, Sink, and Toilet. In this case, we consider 5 different scene types: Living Room, Dining Room, Bedroom, Kitchen, and Bathroom.

In all our tests we use QVGA images (320 × 240 pixels) and a sliding window procedure that considers five different window shapes: square windows, tall rectangular windows (height bigger than width in two different proportions), and wide rectangular windows (width bigger than height in two different proportions). All these window shapes are applied using different window sizes, starting from small windows (12 pixels for the smaller window side) and making them grow by a constant number of pixels (half of the initial side sizes) until the bigger window side is bigger than the corresponding image size. The total number of windows, considering all shapes and scales, is ≈90 000.

We limit our tests to data collected in the previous environments because there is currently a lack of public data sources that contain visual and 3D range information from indoor environments. Thus, as an additional contribution of our paper, we make available a website that contains the code and datasets used in our experiments, we expect this can help further research in the area. This website can be accessed at http://web.ing.puc.cl/~pespinac/ISR.

5.1. Tests using single images

In this sub-section, we run the scene recognition procedure using single images, reporting the most likely scene type where each image was acquired. First, we provide a qualitative evaluation of the proposed method. Then, we compare the performance of our method against 3 alternative approaches. Finally, we provide an evaluation of the improvements produced by using our adaptive object search strategy.

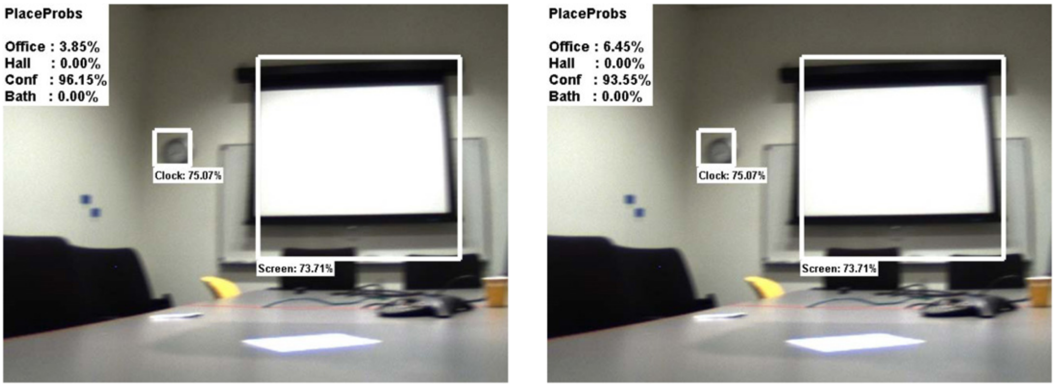


(a) DCC-PUC.

(b) CSAIL-MIT.

(c) Focus of attention for CSAIL-MIT.

Fig. 1. (a)–(b) Executions at two different Office scenes. (c) Focus of attention mechanism applied to image in (b).



(a) Execution 1.

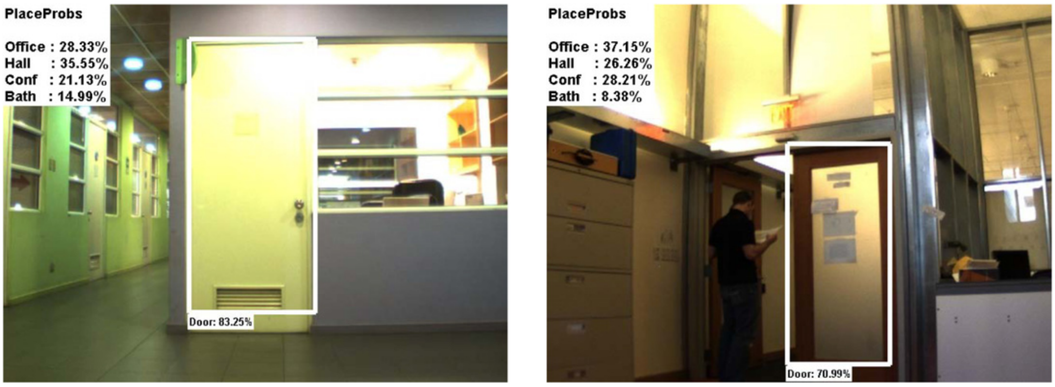
(b) Execution 2.

Fig. 2. Two different executions for the same image in a Conference Room scene. Both executions provide slightly different results with respect to the confidence in place recognition. This is due to the Monte Carlo sampling process.

5.1.1. *Qualitative evaluation of scene recognition*

Fig. 1 shows two different cases where PC-Monitors are detected, at DCC-PUC (Fig. 1(a)) and CSAIL-MIT (Fig. 1(b)). Given that Monitors are more related to Offices than to the rest of the places, Office is the most likely label for the corresponding scenes. We can see that the method makes a good decision when it finds a single object instance (Fig. 1(a)) as well as when it finds more than one instance (Fig. 1(b)). Due to our sliding window procedure, some of the instances are found inside square windows, while others are found inside wide rectangular windows. Additionally, Fig. 1(c) provides a view of the focus of attention mechanism applied to the case of Fig. 1(b). We can see that the method efficiently discards unlikely places, focusing processing in image areas that are highly likely to contain Monitors.

Fig. 2 shows an example image where different executions produce slightly different results. This is due to the sampling procedure. In order to estimate a suitable number of samples, we test our approach using different numbers of samples and we evaluate the variance over identical executions. As expected, increasing the number of samples reduces variance. In our tests, we found that good results can be achieved by using a number of samples in the order of hundreds for each summation in Eq. (14). In particular, in our final implementation we use ≈1000 samples for the external summation and ≈100 for the internal summation in Eq. (14).



(a) Hall is the most likely place.

(b) Office is the most likely place.

Fig. 3. Two different executions where Doors are detected.

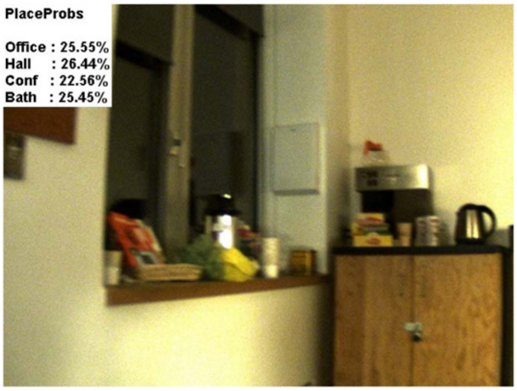


Fig. 4. Example image where no objects are detected.

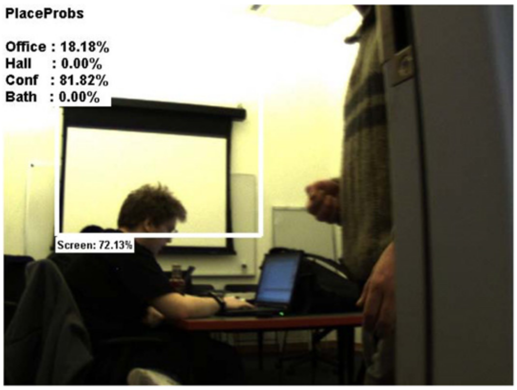


Fig. 5. Detection of a Screen allows our method to correctly label this image as a Conference Room. OT-G and LA-SP methods confuse this scene with an Office due to appearance similarities between Offices and Conference Rooms.

Fig. 3 shows that some objects, such as Doors, cannot discriminate between different places. In this example, both images are taken in Hall scenes. Fig. 3(a) shows an image where a Door is detected and Hall becomes the most likely place, while Fig. 3(b) shows a case where a Door is detected and Office becomes the most likely place. In our experiments, we have found that when only Doors are detected, Hall is slightly more likely than other places, which is consistent with our object–scene priors, $p(Hall|Door) = 0.47$ and $p(Office|Door) = 0.27$. Fig. 4 shows a scenario where no objects are detected, thus, the resulting posterior distribution for place recognition depends only on the scene priors. Here, we assume flat priors for the distribution of scene class, where the posterior probabilities in Fig. 4 show small variations around these priors due to the effect of sampling.

5.1.2. Comparison against alternative approaches

Next, we provide an experimental comparison of our method with respect to 3 alternative approaches: (i) Oliva and Torralba Gist approach (OT-G) [13], which is the same approach used as a baseline in [15], (ii) Lazebnik et al. spatial pyramid approach (LA-SP) [37], and (iii) Sivic et al. pLSA approach applied to scene recognition [36]. We train these models using 100 images for each of the possible scenes. We manually select these images from Google Images with the goal of obtaining a representative set for each scene. In (i) and (ii), we use a SVM classifier with Gaussian radial basis functions as kernels. In (iii), SIFT points are used to build a visual vocabulary and pLSA is used for classification. For all input images, we report the most likely place as the scene detected.

For testing, we mix examples from the 2 Office environments (DCC-PUC and CSAIL-MIT), we use a total number of ≈100 images per class where at least one object is detected. Table 1 shows detection rates (confusion matrices) for our method (OM) and the alternative methods in each of the available scenes. We can see that our method outperforms the alternative approaches by a large margin. In particular, we can see that alternative methods tend to confuse Office and Conference Room, as both places look very alike. Our approach presents good performance for these scenarios, as it can solve ambiguities by detecting discriminative objects, such as a Screen. To support this statement, Fig. 5 shows an example where our method makes a good decision by assigning Conference Room to the underlying scene, despite partial occlusion of the only detected object. In this case, all alternative methods detect the place as Office.

As an additional test, we also trained alternative methods with data coming from the testing environments, this is, images coming from CSAIL-MIT and DCC-PUC, instead of Google Images. We use 100 images for each of the 4 possible scenes, different from the images used for testing. Table 2 shows detection rates for our method and each of the alternative methods in this case.

The previous results indicate that when we train the models with similar amounts of generic data taken from the web, and afterwards, we test each model using images from an independent indoor environment, the proposed method achieves an average recognition rate of 90% while the accuracy of alternative methods ranges around 60%. In a second experiment, when we train the alternative models with images coming from the same Office environment used for testing, we observe more competitive results. In this case, the best performing alternative model is pLSA which reaches an average accuracy of 88% that is closer but still lower than the proposed method trained with generic data. These results demonstrate suitable generalization capabilities of the proposed method and also support previous claims indicating that current state-of-the-art methods for scene recognition present poor performance and low generalization capabilities for the case of indoor scenes.

Table 1
Confusion matrices for compared methods.

Scene	Off. (%)	Hall (%)	Conf. (%)	Bath. (%)	Off. (%)	Hall (%)	Conf. (%)	Bath. (%)
	OM				OT-G			
Office	**91**	7	2	0	**56**	12	26	6
Hall	7	**89**	4	0	13	**52**	15	20
Conference	7	7	**86**	0	72	7	**14**	7
Bathroom	0	6	0	**94**	0	9	15	**76**
	LA-SP				pLSA			
Office	**44**	14	31	11	**64**	11	23	2
Hall	19	**51**	17	13	14	**59**	15	12
Conference	38	16	**41**	5	27	12	**51**	10
Bathroom	2	7	13	**78**	4	9	5	**82**

Table 2
Confusion matrices for compared methods (second test).

Scene	Off. (%)	Hall (%)	Conf. (%)	Bath. (%)	Off. (%)	Hall (%)	Conf. (%)	Bath. (%)
	OM				OT-G			
Office	**91**	7	2	0	**83**	4	13	0
Hall	7	**89**	4	0	7	**86**	5	2
Conference	7	7	**86**	0	17	3	**79**	1
Bathroom	0	6	0	**94**	3	5	3	**89**
	LA-SP				pLSA			
Office	**72**	6	22	0	**88**	4	7	1
Hall	19	**71**	9	1	4	**87**	5	4
Conference	24	6	**67**	3	11	2	**85**	2
Bathroom	2	4	3	**91**	1	4	3	**92**

5.1.3. Tests using adaptive object search

Fig. 6 shows executions of our method in an Office environment at CSAIL-MIT. Fig. 6(a) shows a case without using adaptive object search, while Fig. 6(b) shows a case where we include adaptive object search. In the adaptive case, we add classifiers until the value of the respective information gain is lower than a predefined threshold. We can see that in both cases detections are almost identical, and results differ slightly due to the sampling effect. The main difference between both executions is that in the first case all object detectors are executed, while in the second case the method runs only 5 object detectors: Screen, Urinal, Railing, Soap Dispenser, and Monitor. The reason for this behavior is that at the beginning of the inference process, when no objects are still detected, the adaptive object search scheme chooses to run classifiers associated with objects that are highly discriminative with respect to a specific scene type, such as a Screen or a Urinal, because the eventual detection of those objects maximizes information gain. This is an expected result because we initially use a flat prior for the scene distribution and therefore the detection of informative objects produces peaked posteriors.

In the previous case, by avoiding to run 2 object classifiers, the computational time for the object recognition task is reduced by a factor of ≈1.41. It is important to notice that using adaptive object search also produces an overhead, as the estimation of information gain needs to simulate the scene recognition process for every potential new object type. As we show later, our tests indicate that this overhead is not significant.

Fig. 7 shows executions in a Conference Room environment at DCC-PUC, without using adaptive object search (Fig. 7(a)), and using adaptive object search (Fig. 7(b)). We can see that when adaptive object search is not used, a Screen and a Monitor are detected, while when it is used only the Screen is detected. The reason for this behavior is that at the beginning of the inference process, when no objects are still detected, the object classifiers are executed in the same order as in the previous example, thus, Screen detector is the first classifier to be executed. Given that a Screen is detected with high confidence, the detection of any of the other objects is considered not useful by the information gain metric, thus, the Monitor detector is not executed. In both cases, the final scene recognition is correct and provides a similar posterior distribution. This confirms our intuition that high confidence detections of only a subset of the objects present in the scene is usually sufficient to achieve a correct inference. In this case, by avoiding to run 6 object classifiers, the computational time speed-up for the object recognition task is ≈4.7, where speed-up is defined as the quotient between execution times of the original implementation and the adaptive implementation.

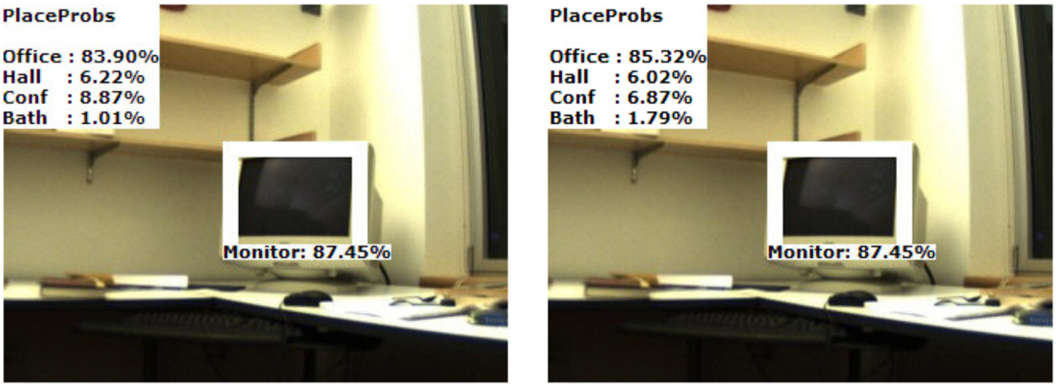
Table 3
Average speed-up using adaptive object search.

Scene	Speed-up
Office	1.92
Hall	1.47
Conference	3.71
Bathroom	2.94
Average	2.51

Fig. 8 shows executions in a Hall environment at CSAIL-MIT, without using adaptive object search (Fig. 8(a)), and using adaptive object search (Fig. 8(b)). We can see that in both cases detections are similar. In the adaptive case after finding instances of Railings no further detections are considered, obtaining processing speed-up of ≈2.12.

Fig. 9 shows a worst-case scenario, where no object is detected, without using adaptive object search (Fig. 4(a)), and using adaptive object search (Fig. 9(b)). When no objects are found, all object detectors are executed, thus, execution time increases instead of being reduced, because of the overhead produced by the adaptive object search procedure. Nevertheless, this example is interesting to analyze, as it shows two additional facts about our adaptive object search scheme: (i) The computational time speed-up is ≈0.94, which confirms that the overhead mentioned earlier is low, (ii) Doors are the last object type to be searched for, which is an expected result because they are the least distinguishing object for the scenes considered in this study.

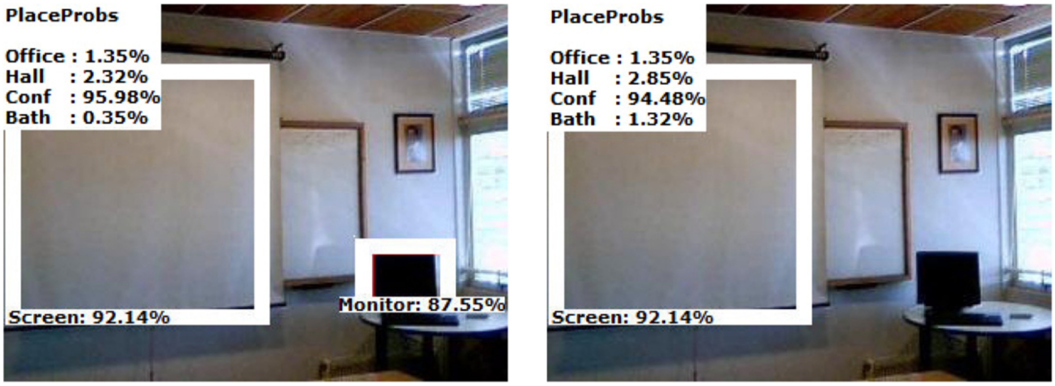
Table 3 shows the average speed-ups produced by using adaptive object search with respect to the case where it is not used, in



(a) Without adaptive object search.

(b) With adaptive object search.

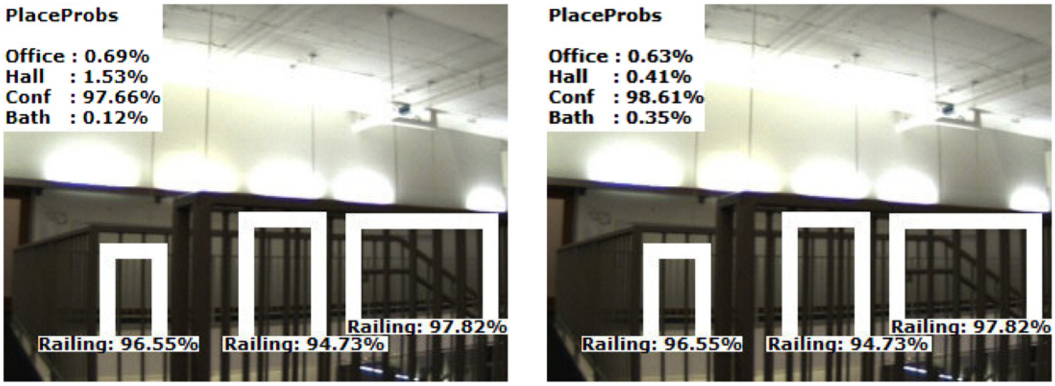
Fig. 6. Executions in an Office environment at CSAIL-MIT.



(a) Without adaptive object search.

(b) With adaptive object search.

Fig. 7. Executions in a Conference Room environment at DCC-PUC.



(a) Without adaptive object search.

(b) With adaptive object search.

Fig. 8. Executions in a Hall environment at CSAIL-MIT.

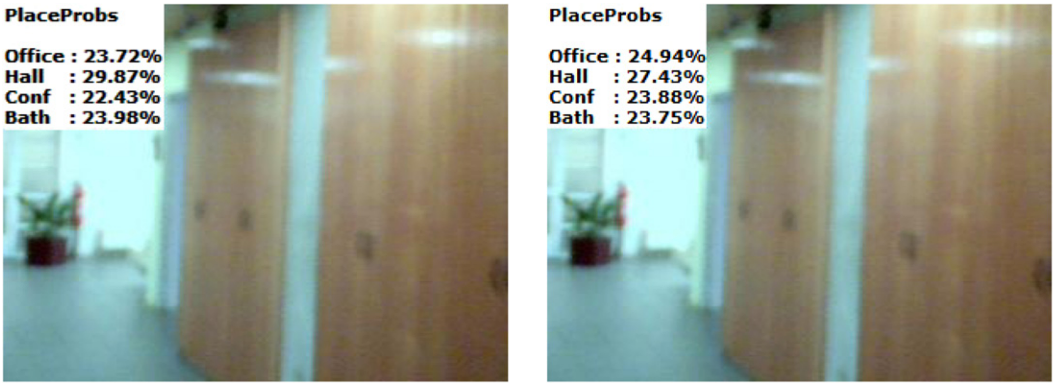
images where objects are found. We can see that these speed-ups differ in the different scene types. While this gain in performance might be considered marginal, the advantage of using an adaptive object search grows with the number of object classifiers available. Therefore, in a large scale case an efficient object search will become a critical tool to avoid the execution of hundreds of object detectors.

5.2. Tests using sequence of images

As we mentioned before, scene recognition is facilitated by the embedded nature of a mobile robot. In this section, we run our scene recognition procedure using sequences of images acquired by a mobile robot during its navigation through Office and home environments. In all tests, we assume that while no objects are detected the scene probability distribution is flat. Also, we consider that an object is present if it is detected at least 8 times over the last 10 frames. This is very useful to avoid false positives that may appear due to noise in some frames.

5.2.1. Office environment

In general, in our test with Office environments the robot is able to correctly recognize the different places using the adaptive and non-adaptive object search. Fig. 10 shows a map of part of DCC-PUC Office environment, displaying the trajectory followed by the robot



(a) Without adaptive object search.

(b) With adaptive object search.

Fig. 9. Executions where no object is detected.

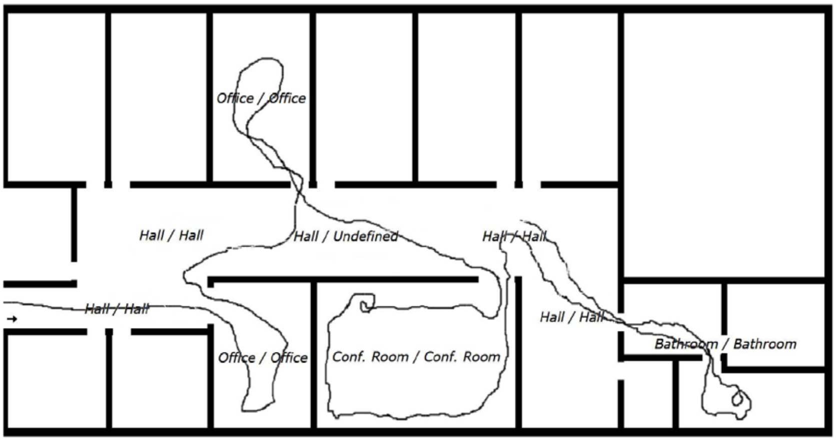


Fig. 10. Map of part of DCC-PUC Office environment, displaying the trajectory followed by the robot during one of its runs. The map also shows the labels assigned by the robot to the visited places, as well as the ground truth label for each place (ground truth/estimation).

during one of its runs. This figure also shows the label estimated by the robot for each of the visited places together with the correct label (ground truth/estimation). We can see that the robot is able to correctly label most of the visited places, with the exception of one of the times when it crosses the Hall, where the place remains undefined. This is due to the fact that during the brief moment when the robot crosses the Hall, it does not detect any object. In particular, the DCC-PUC environment does not have Railings, therefore the only objects detected at Halls are Doors, which are just slightly more related to Halls than to the other places. In this example, the overall average speed-up using an adaptive object search is 2.77, influenced by a relatively fast finding of objects in most of the places, especially in the Conference Room and the Bathroom.

5.2.2. Home environment

In this last experiment, we test the performance of our indoor scene recognition approach using a larger set of object classifiers for the case of a robot wandering in a home environment. As described earlier, we consider 12 object classifiers and 5 types of scenes. Fig. 11 shows a map of this Home environment. The figure also displays the trajectory followed by the robot and the labels of the places visited (ground truth/estimation). Next, we provide insights about the evolution of these scene detections.

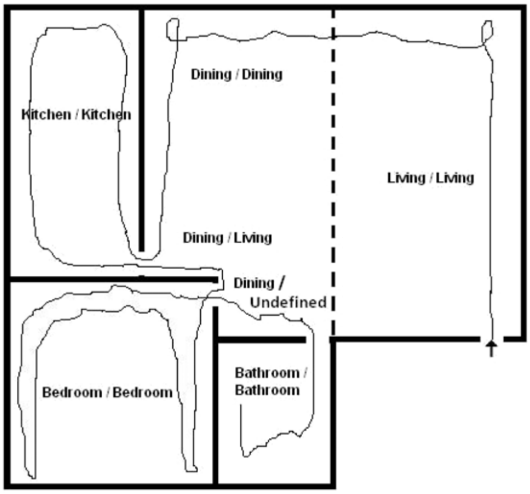
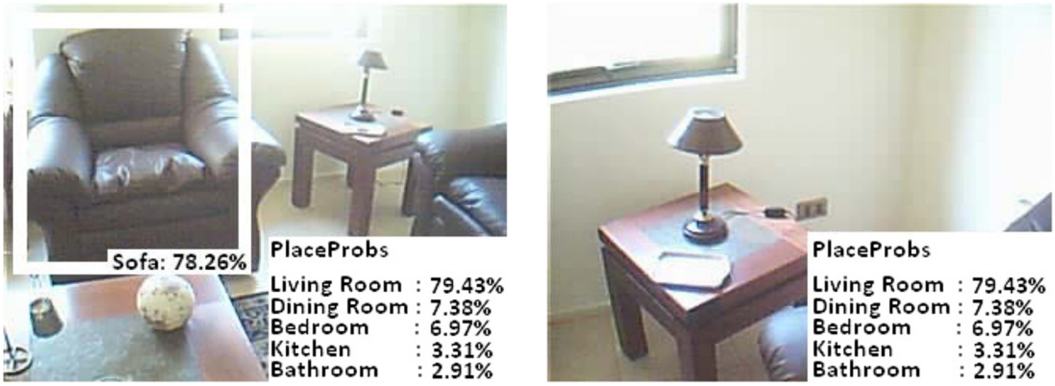


Fig. 11. Map of part of a home environment where the robot navigates, showing the trajectory followed by the robot and the labels of visited places (ground truth/estimation).

First the robot enters a Living Room where a Sofa is detected (Fig. 12(a)). This detection triggers the execution of the scene recognition procedure that increases the likelihood of being in a Living Room. After detecting the Sofa with high confidence, several object classifiers are not executed in the following frames, including the Lamp detector. This explains the missed detection of the Lamp in Fig. 12(b). At this point the robot only runs object detectors corresponding to objects that are highly associated to alternative explanations of the current scene, such as Bed, Toilet, Fridge, and Microwave. It is interesting to note that humans are usually also not aware of all the objects present in a scene. This phenomenon



(a) A Sofa is detected.

(b) A Lamp is not detected.

Fig. 12. Proposed approach operating inside a Living Room in a home environment.

knows as change-blindness has been extensively studied in the literature [52].

Afterwards, the robot enters a Dining Room, which shares a common space with the Living Room (one big room only separated by furniture). This fact is represented by a dotted line on the map in Fig. 11. Here, the robot detects a Chair, and then a Dining Table, recognizing the Dining Room with high confidence.

After leaving the Dining Room, the robot enters a Kitchen. Here, it first finds a Microwave (Fig. 13(a)), which increases the probability of being in a Kitchen. Afterwards, it detects a Fridge (Fig. 13(b)). At this point the robots are highly confident of being in a Kitchen, so a third object is present in the scene, a Sink, is not detected because the corresponding classifier is no longer executed (Fig. 13(c)).

After the Kitchen, the robot crosses the Dining Room in the direction of the Bedroom. Here the robot makes a mistake, as it labels the Dining Room as a Living Room. This is due to the correct detection of a Sofa located in the Living Room that shares a common space with the Dining Room (Fig. 14). In Fig. 14, note also that in the scene there is a Lamp, a Potted Plant, and a second Sofa. These objects are not detected possible due to problems with the lighting conditions and robot pose. This suggests that there are still open research issues to improve the performance of the object detectors.

When the robot enters the Bedroom, it almost immediately finds a Bed, increasing the probability of being in a Bedroom scene (Fig. 15(a)). However, as the Bed detection does not have high confidence, the robot still looks for other objects, such as the TV Monitor that is found afterwards (Fig. 15(b)). Note that in this case the Bedroom probability actually decreases after finding the TV Monitor. The reason for this is that, as the detection of the TV Monitor has high confidence, the probability of being in a scene that also usually has TV Monitors like a Living Room also increases.

After the Bedroom, the robot briefly crosses the Dining Room. In this occasion no object is detected and the scene recognition remains undefined. Finally, the robot enters the Bathroom that is correctly recognized after detecting a Toilet with high confidence.

6. Conclusions and future work

In this work, we present a new hierarchical probabilistic model for indoor scene recognition based on a semantic intermediate representation given by the explicit detection of common objects. As a major finding, we confirm our hypothesis about the advantages of using this type of intermediate representation. In particular, our results indicate that this representation boosts recognition performance, allowing us to overcome some of the limitations of previous methods for the case of indoor scenes. Furthermore, this representation facilitates the acquisition of training data from public websites and also provides a straightforward interpretation of results, for example by identifying failure cases where some relevant objects are not detected. This is in contrast to alternative methods based on generative unsupervised latent models, such as pLSA and DLA, where the intermediate representation does not provide a direct interpretation.

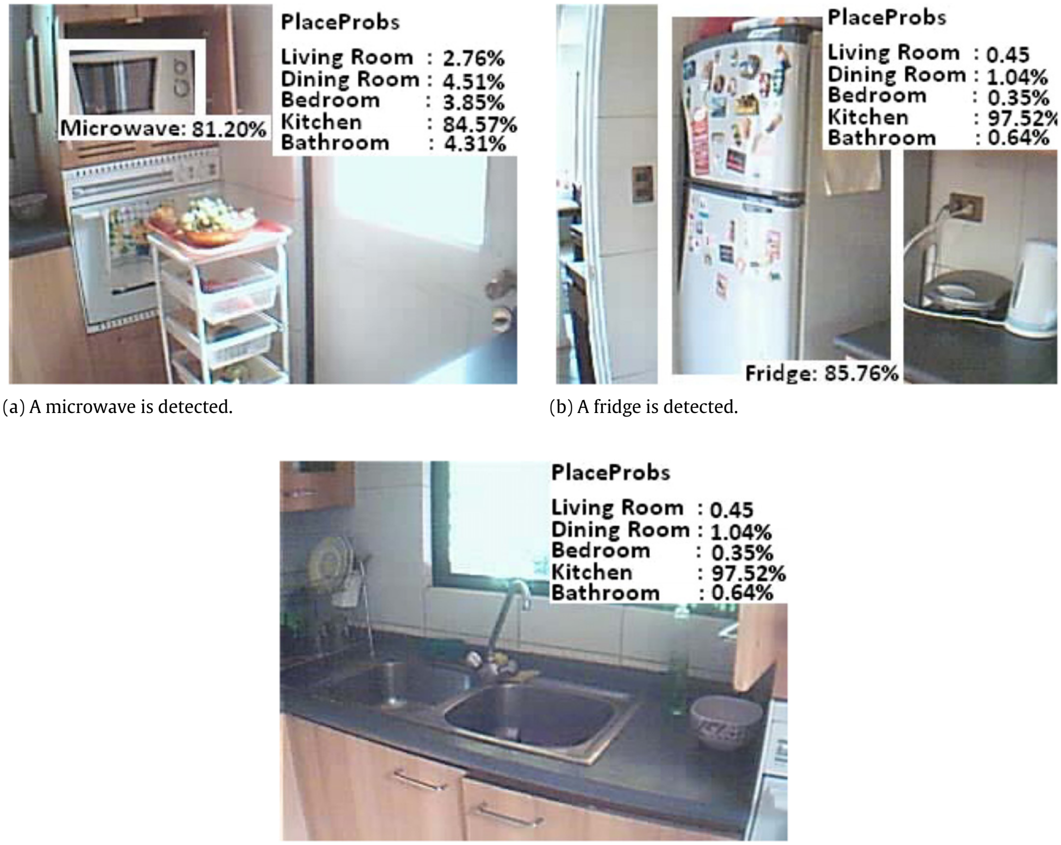
A comparison with alternative methods using images coming from two indoor Office environments indicates that our approach achieves a significant increase in recognition performance. This is true even when the proposed approach is trained using generic data from the web, while the alternative approaches are trained with images coming from the same testing environment. This verifies our initial claim that current methods for scene recognition present poor performance and low generalization capabilities for the case of indoor environments.

It is important to note that our method is only able to make inferences when objects are detected. This assumption is not realistic for the general case of indoor scene recognition, such as the type of algorithms needed for applications like image retrieval, however, we believe that is a valid assumption for the case of an indoor mobile robot that has the chance to explore the environment, capturing a large set of images from a particular place. In fact, in our test using sequences of images, we observe that most of the places are correctly labeled, despite the fact that objects are detected at different times inside each scene.

In terms of object detection, we show the relevance of using reliable 3D information, such as the one provided by 3D range sensors. In our case, the focus of attention mechanism provided by the use of 3D geometrical properties becomes a key element to achieve an efficient sampling scheme to process relevant bounding boxes.

With respect to the use of an adaptive strategy for object search, our tests illustrate the advantages of an efficient on-line selection of object classifiers. In particular, the proposed approach based on maximizing expected information gain presents a desirable behavior. Highly informative objects, such as Screens, are searched first, while uninformative objects, such as Doors, are the last to be searched for. We believe that the relevance of this adaptive scheme can be improved by using more effective priors. This is particularly useful for the case of mobile robotics, where the robot can obtain useful contextual information from its mapping and localization modules. A deeper study of topics related to relevance, use, and sources of more informative priors, is part of our future work.

The processing speed-up provided by our adaptive object search depends on the underlying scene. Some scenes with more distinguishing objects, such as Screen or Urinal, display higher average speed-ups when these objects are quickly detected. Scenes



(a) A microwave is detected.

(b) A fridge is detected.

(c) A sink is not detected.

Fig. 13. Proposed approach operating inside a Kitchen in a home environment.

with less distinguishing objects, such as Doors, display lower average speed-ups. As we mentioned, the advantage of using an adaptive object search grows with the number of object classifiers available, therefore, we believe that this scheme is a fundamental element for a large scale implementation of the proposed method, where an efficient object search can avoid the execution of hundreds of object detectors.

In terms of processing time, depending on the number of windows that are discarded at early stages of the object detection, our (non-optimized) implementation running on a regular laptop computer currently takes in the order of 20 s to process each image. Results show that by using adaptive object search we can reduce this execution time by a factor or 2 or 4, but we still do not have a real time implementation. Given that our method is highly parallelizable, we believe that an efficient implementation can run in real time, for example using GPU hardware. A real time version of the proposed method is part of our current efforts.

Among failure cases, the proposed approach is not able to label scenes where objects are not detected or are detected with low confidence. Also, we observe detection problems in cases where rare object combinations are found. As an example, in one of our tests a Monitor is detected in a Hall environment, and only Doors are detected next to it. As a result the place is labeled as an Office. These previous failure cases highlight two limitations of our current approach. First, images where no objects are detected cannot be identified. In this sense, we are currently providing our robot with active perceptual behaviors [53] that can guide its motions in order to find suitable views of key objects. A second and related limitation arises from the assumption that contextual relations among object combinations and places can be inferred from databases such as Flickr. For single images this assumption is reasonable, however, it becomes problematic when many objects are detected inside a particular place, as single images extracted from Flickr usually do not have all these objects together. This results in long object combinations receiving an almost null probability. In this work, this was not a relevant issue because we did not use many object detectors, however, a more extensive implementation with a large amount of object detectors will need to consider this case. A straightforward solution is to use training data to build probabilistic factorizations of long joint combinations of objects. This idea is also part of our future work.

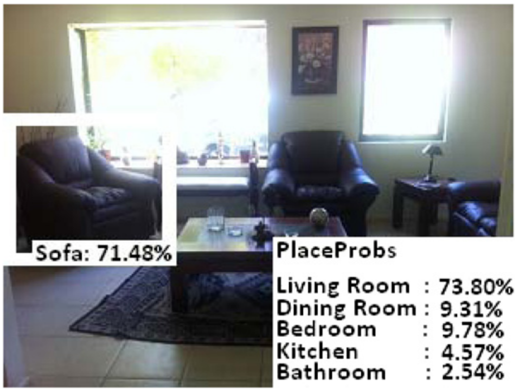
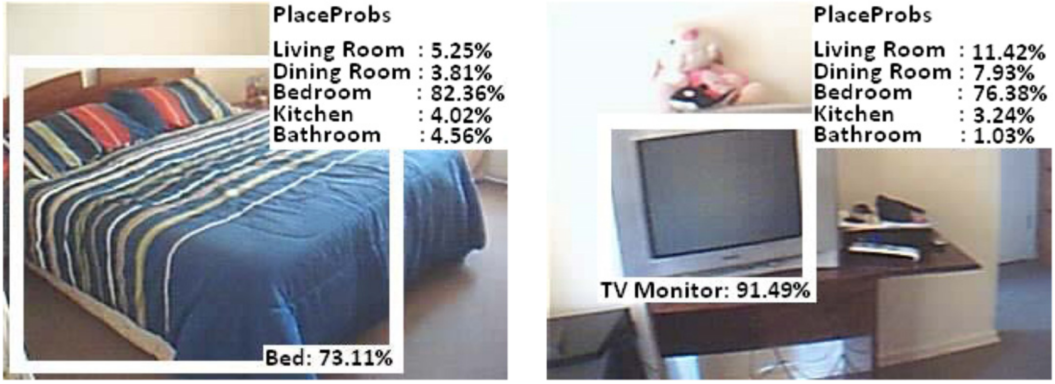


Fig. 14. A Sofa detected in a Living Room, but seeing when the robot is actually in a Dining Room. This produces an error in the scene recognition.

Acknowledgment

This work was partially funded by FONDECYT grant 1095140.



(a) A Bed is detected.

(b) A TV Monitor is detected.

Fig. 15. Method operating inside a Kitchen.

References

[1] iRobot, http://store.irobot.com/, 2012.

[2] Probotics, http://www.probotics.com/, 2012.

[3] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, Cambridge University Press, New York, 2006.

[4] C. Siagian, L. Itti, Rapid biologically-inspired scene classification using features shared with visual attention, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2) (2007) 300–312.

[5] P. Espinace, D. Langdon, A. Soto, Unsupervised identification of useful visual landmarks using multiple segmentations and top-down feedback, Robotics and Autonomous Systems 56 (6) (2008) 538–548.

[6] E. Brunskill, T. Kollar, N. Roy, Topological mapping using spectral clustering and classification, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2007, pp. 3491–3496.

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

[8] P. Viola, M. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 137–154.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2013).

[10] T. Collett, Landmark learning and guidance in insects, Proceedings of the Royal Society of London, Series B (1992) 295–303.

[11] D.H. Ballard, Animate vision, Artificial Intelligence 48 (1991) 57–86.

[12] I. Ulrich, I. Nourbakhsh, Appearance-based place recognition for topological localization, in: IEEE International Conference on Robotics and Automation, 2000.

[13] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, International Journal of Computer Vision 42 (2001) 145–175.

[14] L. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2005.

[15] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2009.

[16] T. Hofmann, Probabilistic latent semantic analysis, in: Uncertainty in Artificial Intelligence, 1999.

[17] D. Blei, A. Ng, M. Jordan, Latent Dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.

[18] C. Fredembach, M. Schroder, S. Susstrunk, Eigenregions for image classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (12) (2004) 1645–1649.

[19] A. Mojsilovic, J. Gomes, B. Rogowitz, Isee: perceptual features for image library navigation, in: SPIE Human Vision and Electronic Imaging, 2002.

[20] T. Kollar, N. Roy, Utilizing object–object and object–scene context when planning to find things, in: International Conference on Robotics and Automation, 2009.

[21] P. Dollar, Z. Tu, P. Perona, S. Belongie, Integral channel features, in: British Machine Vision Conference, 2009.

[22] S. Vasudevan, R. Siegwart, Bayesian space conceptualization and place classification for semantic maps in mobile robotics, Robotics and Autonomous Systems 56 (2008) 522–537.

[23] P. Espinace, T. Kollar, A. Soto, N. Roy, Indoor scene recognition through object detection, in: IEEE International Conference on Robotics and Automation, 2010.

[24] P. Viswanathan, T. Southey, J. Little, A. Mackworth, Automated place classification using object detection, in: Canadian Conference on Computer and Robot Vision, 2010.

[25] L.-J. Li, H. Su, E. Xing, L. Fei-Fei, Object bank: a high-level image representation for scene classification and semantic feature sparsification, in: Neural Information Processing Systems, 2010.

[26] S. Thorpe, C. Fize, C. Marlot, Speed of processing in the human visual system, Nature 381 (1996) 520–522.

[27] A. Bosch, X. Muñoz, R. Martí, A review: which is the best way to organize/classify images by content? Image and Vision Computing 25 (2007) 778–791.

[28] A. Vailaya, A. Jain, H. Zhang, On image classification: city vs. landscapes, Pattern Recognition 31 (1998) 1921–1935.

[29] A. Vailaya, M. Figueiredo, A. Jain, H. Zhang, Content-based hierarchical classification of vacation images, in: IEEE Int. Conf. on Multimedia Computing and Systems, 1999.

[30] E. Chang, K. Goh, G. Sychay, G. Wu, Cbsa: content-based soft annotation for multimodal image retrieval using Bayes point machines, IEEE Transactions on Circuits and Systems for Video Technology 13 (2003) 26–38.

[31] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (11) (1998) 1254–1259.

[32] M. Szummer, R. Picard, Indoor–outdoor image classification, in: IEEE International Conference on Computer Vision, Workshop on Content-based Access of Image and Video Databases, 1998.

[33] S. Paek, S. Chang, A knowledge engineering approach for image classification based on probabilistic reasoning systems, in: IEEE International Conference on Multimedia and Expo, 2000.

[34] J. Vogel, B. Schiele, A semantic typicality measure for natural scene categorization, in: DAGM Pattern Recognition Symposium, 2004.

[35] A. Bosch, A. Zisserman, X. Muñoz, Scene classification via pLSA, in: European Conference on Computer Vision, 2006.

[36] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, W.T. Freeman, Discovering objects and their localization in images, in: International Conference in Computer Vision, 2005.

[37] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2006.

[38] A. Pronobis, O. Mozos, B. Caputo, P. Jens-felt, Multi-modal semantic place classification, The International Journal of Robotics Research 29 (2–3) (2010) 298–320.

[39] A. Pronobis, Semantic mapping with mobile robots, Ph.D. Thesis, Royal Institute of Technology, Stockholm, Sweden, 2011.

[40] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: European Conf. on Comp. Vision, 2005.

[41] P. Felzenszwalb, R. Girshick, D. McAllester, Cascade object detection with deformable part models, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2010.

[42] S. Helmer, D. Lowe, Using stereo for object recognition, in: IEEE International Conference on Robotics and Automation, 2010.

[43] M. Cummins, P. Newman, FAB-MAP: probabilistic localization and mapping in the space of appearance, International Journal of Robotics Research 27 (6) (2008) 647–665.

[44] B. Douillard, D. Fox, F. Ramos, Laser and vision based outdoor object mapping, in: Robotics: Science and Systems, 2008.

[45] I. Posner, M. Cummins, P. Newman, A generative framework for fast urban labeling using spatial and temporal context, Autonomous Robots 26 (2–3) (2009) 153–170.

[46] B. Russell, A. Torralba, K. Murphy, K. Freeman, Labelme: a database and web-based tool for image annotation, International Journal of Computer Vision 77 (1–3) (2008) 157–173.

[47] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, in: IEEE International Conference on Computer Vision and Pattern Recognition, Workshop on Generative-Model Based Vision, 2004.

[48] A. Bosch, A. Zisserman, X. Muñoz, Image classification using random forests and ferns, in: IEEE International Conference on Computer Vision, 2007.

[49] S. Montabone, A. Soto, Human detection using a mobile platform and novel features derived from a visual saliency mechanism, Image and Vision Computing 28 (3) (2010) 391–402.

[50] Y. Freund, R. Schapire, A decision–theoretic generalization of on-line learning and an application to boosting, Computer and System Sciences 55 (1) (1997) 119–139.
[51] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.
[52] D. Simons, D. Levin, Change blindness, Trends in Cognitive Sciences 1 (7) (1997) 261–267.
[53] J. Correa, A. Soto, Active visual perception for mobile robot localization, Journal of Intelligent and Robotic Systems 58 (3–4) (2010) 339–354.

Pablo Espinace is a Research Scientist in the Computer Science Department at Pontificia Universidad Catolica de Chile. He received his M.Sc. degree and Ph.D. in Computer Science from Pontificia Universidad Catolica de Chile in 2011. His research interests include Mobile Robotics, Machine Learning and Computer Vision.

Thomas Kollar has a Ph.D. in Electrical Engineering and Computer Science (EECS) from the Massachusetts Institute of Technology (MIT). His thesis concerned learning to understand spatial natural language commands and his research interests include robot learning, language grounding, and human–robot interaction. He was the general chair of the HRI Pioneers Workshop at the 6th ACM/IEEE International Conference on Human–Robot Interaction. He received his Master of Science in EECS from MIT in 2007 for research in reinforcement learning towards improving the quality of robot mapping. He has a Bachelor of Science in Computer Science (with Honors) and a Bachelor of the Arts in Mathematics from the University of Rochester. As an undergraduate, he developed an hors d'oeuvre-serving robot as a part of the AAAI robotics competition and is a member of IEEE, AAAI, and Sigma Xi and has published at HRI, ISER, ICRA, AAAI, IROS and ICMI. He is currently a Postdoctoral Fellow in the Computer Science Department at Carnegie Mellon University.

Nicholas Roy is an Associate Professor in the Department of Aeronautics & Astronautics at the Massachusetts Institute of Technology and a member of the Computer Science and Artificial Intelligence Laboratory at MIT. He received his Ph.D. in Robotics from Carnegie Mellon University in 2003. His research interests include mobile robotics, decision-making under uncertainty, human–computer interaction, and machine learning.

Alvaro Soto received his Ph.D. in Computer Science from Carnegie Mellon University in 2002; and a M.Sc. degree in Electrical and Computer Engineering from Louisiana State University in 1997. He joined the Computer Science Department at Pontificia Universidad Catolica de Chile, where he became an Associate Professor in 2007. His main research interests are in Statistical Machine Learning and Mobile Robotics.