Towards Semi-Supervised Learning of Semantic Spatial Concepts for Mobile Robots

Jesus Martinez-Gomez and Barbara Caputo

***Abstract*—The ability of building robust semantic space representations of environments is crucial for the development of truly autonomous robots. This task, inherently connected with cognition, is traditionally achieved by training the robot with a supervised learning phase. We argue that the design of robust and autonomous systems would greatly benefit from adopting a semi-supervised online learning approach. Indeed, the support of open-ended, lifelong learning is fundamental in order to cope with the dazzling variability of the real world, and online learning provides precisely this kind of ability. Here we focus on the robot place recognition problem, and we present an online place classification algorithm that is able to detect gap in its own knowledge based on a confidence measure. For every incoming new image frame, the method is able to decide if (a) it is a known room with a familiar appearance, (b) it is a known room with a challenging appearance, or (c) it is a new, unknown room. Experiments on ImageCLEF database and a subset of the challenging COLD database show the promise of our approach.**

***Index Terms*—place recognition, semantic place representation, online learning, kernel methods.**

I. Introduction

WHO wouldn't want a robot at home to make the daily chores? It could bring you a beer from the fridge, do the laundry, iron the shirts, collect things from the floor before cleaning, etc. A major requirement for having robots at home is that their representation of space, objects, and more generally concepts must at least partially overlap with our own. A vast literature in cognitive psychology (see [1] and reference therein) shows clearly that humans explain and categorize perceived multi-sensory patterns using semantic representations, of which language represents the synthesis. To fix ideas, let us focus here only on the semantic representation of space. We refer to rooms, and talk about them, in terms of their visual appearance (the corridor), the activities we usually perform in them (the fitness room) and the objects they contain (the bedroom). If we want to share our daily environment with robots, we need to share with them our own representation and understanding of it.

How do we make a robot learn the typical semantic space representation of humans? Robots have perceptual channels and cognitive abilities very different from our own. For instance, the typical service robot will use laser range scanners and an omnidirectional camera to collect data about an indoor place like an office environment. If programmed to learn the environment autonomously, i.e., in an unsupervised manner, the robot's interpretation of the data will result in a space representation very different from that of humans. Therefore, to make a robot have our own semantic representation of space, it is necessary to have a learning phase supervised by the user.

But how long should this supervised learning phase be? The current mainstream approaches (see Section II for a brief review of the relevant literature) assume a training phase, well separated from the actual working of the robot, where the human labels the data. Training usually stops when it is achieved a pre-defined threshold level of performance on a validation set of data, or when the user decides it. From that moment on, the robot is on its own. We argue that this approach is doomed to fail: rooms change around us continuously over time as furniture is added, replaced or relocated. It is impossible to predict how a user is going to redecorate its living room in the future, and therefore it is impossible to train the robot beforehand on such data.

Our vision is that the supervised learning mode should always be accessible to the robot, and it should be triggered by its ability to explain the incoming data. The transition from fully supervised to unsupervised should be smooth, robot driven, and competence-based. In other words, our vision is that semi-supervised online learning should become the mainstream approach for enabling robots to learn semantic concepts.

To move towards this goal, here we present an algorithm able to learn semantic spatial concepts in an open ended fashion, i.e. continuously updating its internal model with a bounded memory growth. The robot switches from a fully autonomous, unsupervised learning phase to a supervised one (where assistance by a human teacher might be required) on the basis of its capability to interpret the data with a high degree of confidence. The capability to detect hard-to-explain incoming data is done at the classifier level, frame by frame, as well as at a higher level, by exploiting the temporal continuity of the image sequences. This permits to distinguish between challenging instances of a known spatial concept (a view of the known class kitchen where it is perceived for the first time a new piece of furniture) and a new concept (a room never seen before).

Concretely, our algorithm consists of two components: the first is an online learning algorithm with performance comparable to that of the batch method and a bounded memory growth; the second is a mechanism for assigning labels to incoming data, detecting challenging frames imaging known

Jesus Martinez-Gomez is with the I3A Research Institute, Campus Universitario s/n, 02071, Albacete, Spain. E-mail: jesus_martinez@dsi.uclm.es

Barbara Caputo is with the Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland. E-mail: bcaputo@idiap.ch

This work was supported by the Spanish "Junta de Comunidades de Castilla-La Mancha" under PCI08-0048-8577 and PBI-0210-7127 projects (J. M.-G.) and by the SS2Rob project (B.C.). The support is gratefully acknowledged.

concepts and ultimately recognizing when being in a whole new room. We take an discriminative approach and we build on previous work on online learning [2], [3] and confidence-based place classification [4]. Experiments on a subset of the challenging COLD database [5] and on the database used for the Robot vision Task at the ImageCLEF 2010 challenge evaluation (www.imageclef.org) show promising results.

The rest of the paper is organized as follows: after a brief review of the related literature, we describe the two components of our approach: the online learning algorithm (Section III) and the detection of confidence/ignorance (Section IV). Section VI describes our experimental setup, while section VII reports our experimental findings. We conclude with an overall discussion and possible future avenues for research.

II. Related Works

The ability to learn and interpret complex sensory information based on previous experience, inherently connected with cognition, has been recognized as crucial and vastly researched [6], [7], [8]. In most cases, the recognition systems used are trained offline, i.e., they are based on batch learning algorithms. However, in the real dynamic world, learning cannot be a single act. It is simply not possible to create a static model which could explain all the variability observed over time. Continuous information acquisition and exchange, coupled with an ongoing learning process, is necessary to provide a cognitive system with a valid world representation.

In the last few years, the need for solutions to such problems as the robustness to long-term dynamic variations, or the transfer of knowledge, is more and more acknowledged. In [7], the authors tried to deal with long-term visual variations in indoor environments by combining information acquired using two sensors of different characteristics. In [9], the problem of invariance to seasonal changes in appearance of an outdoor environment is addressed. Clearly, adaptability is a desirable property of a recognition system. At the same time, Thrun and Mitchell [10], [11] studied the issue of exchanging knowledge related to different tasks in the context of artificial neural networks and argued for the importance of knowledge-transfer schemes for lifelong robot learning. Several attempts to solve the problem have also been made from the perspective of Reinforcement Learning, including the case of transferring learned skills between different RL agents [12], [13].

III. Step 1: Memory Controlled Online Learning and Recognition of Visual Places

This section describes the first component of our overall approach, namely an online learning algorithm with a bounded memory growth and an accuracy comparable to the classic, off-line method. We take a discriminative approach, and derive an approximate version of the Online Independent-SVM. As opposed to the original algorithm, our approach does not require to store all incoming data but it allows to discard most of them in a principled manner. This leads to a bounded memory growth, where the upper bound is set by the user and the lower bound by theoretical constraints. In the rest of this section we first review basic concepts on SVM (section III-A), then we summarize the OI-SVM algorithm (section III-B). Our Memory Controlled OI-SVM is described in section III-C.

A. Support Vector Machines

Due to space limitations, this is a very quick account of SVMs — the interested reader is referred to [14] for a tutorial, and to [15] for a comprehensive introduction to the subject. Assume $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, with $\mathbf{x}_i \in \mathbb{R}^m$ and $y_i \in \{-1, 1\}$, is a set of samples and labels drawn from an unknown probability distribution; we want to find a function $f(\mathbf{x})$ such that $sign(f(\mathbf{x}))$ best determines the category of any future sample $\mathbf{x}$. In the most general setting,

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{1}$$

where $b \in \mathbb{R}$ and $K(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1) \cdot \Phi(\mathbf{x}_2)$, the kernel function, evaluates inner products between images of the samples through a non-linear mapping Φ. The α_is are Lagrangian coefficients obtained by solving (the dual Lagrangian form of) the problem

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l}\xi_i^p \\ \text{subject to} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{2}$$

where $\mathbf{w}$ defines a separating hyperplane in the feature space, i.e., the space where Φ lives, whereas $\xi_i \in \mathbb{R}$ are slack variables, $C \in \mathbb{R}^+$ is an error penalty coefficient and p is usually 1 or 2. In practice, most of the α_i are found to be zero after training; the vectors with an associated α_i different from zero are called support vectors. Notice that, from (1), the testing time of a new point is proportional to the number of SVs, hence reducing the number of SVs implies reducing the testing time.

B. Online Independent Support Vector Machines

Let the *kernel matrix* K be defined such that $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, with $i, j = 1, \ldots, l$. The possibility to obtain a more compact representation of $f(\mathbf{x})$ follows from the fact that the solution to a SVM problem (that is, the α_is) is not unique if K does not have full rank [14], which is equivalent to some of the SVs being linearly dependent on some others in the feature space [16]. Orabona et al [2] applied this idea to the online learning framework. As it would be unfeasible a simplification of the solution each time a new sample is acquired, they suggested to use independent SVs only, that is to decouple the concept of "basis" vectors, used to build the classification function (1), from the samples used to evaluate the ξ_i in (2). If the selected basis vectors span the same subspace as the *whole sample set*, the solution found will be equivalent.

The OI-SVM algorithm adds incrementally a new incoming sample if it is linearly independent in the feature space from those already present in the basis itself. The solution found is

the same as in the classical SVM formulation; therefore, no approximation whatsoever is involved.

Denoting the indexes of the vectors in the current basis, after l training samples, by $\mathcal{B}$, and the new sample under judgment by $\mathbf{x}_{l+1}$, the algorithm can then be summed up as follows:

1) check whether $\mathbf{x}_{l+1}$ is linearly independent from the basis in the feature space; if it is, add it to $\mathcal{B}$; otherwise, leave $\mathcal{B}$ unchanged.
2) incrementally re-train the machine.

Hence the testing time for a new point will be $O(|\mathcal{B}|)$, as opposed to $O(l)$ in the standard approach; therefore, keeping $\mathcal{B}$ small will improve the testing time without losing any precision whatsoever. A major drawback of OI-SVM is that it requires to store in memory all the incoming training data in order to guarantee that the online solution is the same as in the classical SVM formulation.

In the following, the notations A_{IJ} and $\mathbf{v}_I$, where A is a matrix, $\mathbf{v}$ is a vector and $I, J \subset \mathbb{N}$ denote in turn the sub-matrix and the sub-vector obtained from A and $\mathbf{v}$ by taking the indexes in I and J.

Linear independence In general, checking whether a matrix has full rank is done via some decomposition, or by looking at the eigenvalues of the matrix; but here one wants to check whether a *single* vector is linearly independent from a matrix which is already known to be full-rank. Inspired by the definition of linear independence, the algorithm checks how well the vector can be approximated by a linear combination of the vectors in the set [17]. Let $d_j \in \mathbb{R}$; then let

$$\Delta = \min_{\mathbf{d}} \left|\left| \sum_{j \in \mathcal{B}} d_j \phi(\mathbf{x}_j) - \phi(\mathbf{x}_{l+1}) \right|\right|^2 \tag{3}$$

If $\Delta > 0$ then $\mathbf{x}_{l+1}$ is linearly independent with respect to the basis, and $\{l+1\}$ is added to $\mathcal{B}$. In practice, it checks whether $\Delta \leq \eta$ where $\eta > 0$ is a tolerance factor, and expects that larger values of η lead to worse accuracy, but also to smaller bases. As a matter of fact, if η is set at machine precision, OISVMs retain the exact accuracy of SVMs. Notice also that if the feature space has finite dimension n, then no more than n linearly independent vectors can be found, and $\mathcal{B}$ will never contain more than n vectors.

Expanding equation (3) one gets

$$\Delta = \min_{\mathbf{d}} (\sum_{i,j \in \mathcal{B}} d_j d_i \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_i)$$

$$-2 \sum_{j \in \mathcal{B}} d_j \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_{l+1}) + \phi(\mathbf{x}_{l+1}) \cdot \phi(\mathbf{x}_{l+1})) \tag{4}$$

that is, applying the kernel trick,

$$\Delta = \min_{\mathbf{d}} \left(\mathbf{d}^T K_{\mathcal{B}\mathcal{B}} \mathbf{d} - 2\mathbf{d}^T \mathbf{k} + K(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) \right) \tag{5}$$

where $k_i = K(\mathbf{x}_i, \mathbf{x}_{l+1})$ with $i \in \mathcal{B}$. Solving (5), that is, applying the extremum conditions with respect to $\mathbf{d}$, one obtains

$$\tilde{\mathbf{d}} = K_{\mathcal{B}\mathcal{B}}^{-1} \mathbf{k} \tag{6}$$

and, by replacing (6) in (5) once,

$$\Delta = K(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) - \mathbf{k}^T \tilde{\mathbf{d}} \tag{7}$$

Note that $K_{\mathcal{B}\mathcal{B}}$ can be safely inverted since, by incremental construction, it is full-rank. An efficient way to do it, exploiting the incremental nature of the approach, is that of updating it recursively: after the addition of a new sample, the new $K_{\mathcal{B}\mathcal{B}}^{-1}$ then becomes

$$\begin{bmatrix} & & & 0 \\ & K_{\mathcal{B}\mathcal{B}}^{-1} & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} + \frac{1}{\Delta} \begin{bmatrix} \tilde{\mathbf{d}} \\ -1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{d}}^T & -1 \end{bmatrix} \tag{8}$$

where $\tilde{\mathbf{d}}$ and Δ are already evaluated during the test (this method matches the one used in Cauwenberghs and Poggio's incremental algorithm [18]). Thanks to this incremental evaluation, the time complexity of the linear independence check is $O(|\mathcal{B}|^2)$, as one can easily see from Equation (6).

With this method OI-SVM approximates the original kernel matrix K with another matrix $\widehat{K}$ [19]; the quality of the approximation depends on η. In fact it is possible to show that $trace(K - \widehat{K}) \leq \eta |\mathcal{B}| \leq \eta l$, where l is the number of samples acquired [20]. If one considers a normalized kernel, that is a kernel for which $K(x, x)$ is always equal to 1, we can write $trace(K - \widehat{K})/trace(K) \leq \eta$. On the other hand a bigger η means of course a smaller number of SVs, hence it controls the trade-off between accuracy and speed of the OISVM.

Training the machine The training method largely follows Keerthi et al. [21], [22], adapted for online training. The algorithm directly minimizes problem (2) as opposed to the standard way of minimizing its dual Lagrangian form, allowing to select explicitly the basis vectors to use. OI-SVM sets $p = 2$ in (2) and transform it to an unconstrained problem. Let $\mathcal{D} \subset \{1, \ldots, l\}$; then the unconstrained problem is

$$\min_{\boldsymbol{\beta}} \left(\frac{1}{2} \boldsymbol{\beta}^T K_{\mathcal{D}\mathcal{D}} \boldsymbol{\beta} + \frac{1}{2} C \sum_{i=1}^{l} max \left(0, 1 - y_i K_{i\mathcal{D}} \boldsymbol{\beta}\right)^2 \right) \tag{9}$$

where $\boldsymbol{\beta}$ is the vector of the Lagrangian coefficients involved in $f(\mathbf{x})$, analogously to the α_is in the original formulation. If one sets $\mathcal{D} = \mathcal{B}$, then the solution to the problem is unique since $K_{\mathcal{B}\mathcal{B}}$ is full rank by construction. Newton's method as modified by Keerthi et al. [21], [22] can then be used to solve (9) after each new sample. When the new sample $\mathbf{x}_{l+1}$ is received the method goes as follows:

1) let $\mathcal{I} = \{i : 1 - y_i o_i > 0\}$ where $o_i = K_{i\mathcal{B}} \boldsymbol{\beta}$ and $\boldsymbol{\beta}$ is the vector of optimal coefficients with l training samples; if $\mathcal{I}$ has not changed, stop.
2) otherwise, let the new $\boldsymbol{\beta}$ be $\boldsymbol{\beta} - \gamma \mathbf{P}^{-1} \mathbf{g}$, where $\mathbf{P} = K_{\mathcal{B}\mathcal{B}} + C K_{\mathcal{B}\mathcal{I}} K_{\mathcal{B}\mathcal{I}}^T$ and $\mathbf{g} = K_{\mathcal{B}\mathcal{B}} \boldsymbol{\beta} - C K_{\mathcal{B}\mathcal{I}} (\mathbf{y}_{\mathcal{I}} - \mathbf{o}_{\mathcal{I}})$.
3) go back to Step 1.

In Step 2 above, γ is set to one. In order to speed up the algorithm, OI-SVM maintains an updated Cholesky decomposition of $\mathbf{P}$. It turns out that the algorithm converges

in very few iterations, usually 0 to 2; the time complexity of the re-training step is $O(|\mathcal{B}|l)$, as well as its space complexity; hence, keeping $\mathcal{B}$ small will speed up the training time as well as the testing time.

C. Memory Controlled Online Independent Support Vector Machines

The need to store all incoming data makes in practice unusable the OI-SVM algorithm for open-ended learning of semantic spatial concepts, especially for a mobile robot platform: while the dimension of the solution would remain constant over time, the overall memory requirement would grow linearly with the number of perceived frames, leading quickly to a memory explosion.

To overcome this problem, we propose to apply a forgetting strategy over the stored Training Samples (TSs), while preserving the stored Support Vectors in order to approximate reasonably well the original optimal solution. The idea of keeping under control the memory growth of online learning algorithms is not new: several authors tried in the past to address this problem, mainly by bounding a priori the memory requirements. The first algorithm to overcome the unlimited growth of the support set was proposed by Crammer et al. [23]. The algorithm was then refined by Weston et al. [24]. The idea of the algorithm was to discard a vector of the solution, once the maximum dimension has been reached. The strategy was purely heuristic and no mistake bounds were given. A similar strategy has been used also in NORMA [25] and SILK [26]. The very first online algorithm to have a fixed memory "budget" and at the same time to have a relative mistake bound has been the Forgetron [27]. Within the context of semantic scene recognition, Ullah et al [28] proposed instead a random forgetting strategies, which should be more robust to possible unbalancing into the class-by class distribution of the TSs.

Here we take the approach proposed in [28] and define the following random forgetting strategy:

1) we introduce a threshold value that corresponds to the allowed maximum number of stored Training Samples ($MaxTSs$);
2) whenever $TS > MaxTSs$, we randomly discard TSs until their value is again below threshold. This concretely means discarding old TSs, selected randomly, for each new incoming TS.
3) With this strategy, the memory requirements of the algorithm are always between the number of SVs of the testing solution and the number of SVs plus $MaxTSs$.

We will show experimentally that this approximation of the original OI-SVM algorithm does not affect the accuracy of the solution for a wide range of values of $MaxTSs$.

IV. Step 2: Detection of Ignorance

The second, key component of our method is the capability to autonomously assign labels to new, incoming images, without the need for human supervision. The core issue here is the ability to estimate the level of confidence of each potential decision: a frame classified as corridor should be used to update the internal representation for the class corridor only if the confidence of the decision is high enough. If this would not be the case, then there would be a very strong risk of adding wrongly labeled data to the model, with a consequent degradation of the overall performance over time.

At the same time, one could argue that the most challenging frames, for each known class, are the most important to be added as they are those bringing new valuable information. An obvious way to do so would be to store the challenging frames and then, periodically, asking for labels to a human supervisor. Our solution here is instead to exploit the temporal continuity between frames and the intrinsic constraints of the problem: once a robot has traversed a door, all frames perceived until crossing another door must belong to the same semantic spatial concept. This same line of reasoning gives us a useful tool to determine if the robot has entered a new, unknown room.

Section IV-A describes how we estimate the level of confidence of the classifier and how to exploit temporal continuity to label challenging frames. Section IV-B illustrates how these two ingredients can be also used to identify new semantic spatial concepts. Finally, Section IV-C shows how challeging frames can be used to improve future classifications without having identified a new room.

A. Detecting Challenging Frames

To use incoming data to update the internal models, the algorithm needs to assign reliably class labels to each new frame. This in turns means that it should be able to detect frames that cannot be properly classified, i.e. that cannot be classified with a high confidence level. The problem therefore becomes that of defining effective confidence measures for evaluating the reliability of the label assignement process.

This issue has been widely studied in the literature [29]. Here we follow the classic approach proposed by Platt [30], which turns decision margins into conditional probabilities:

$$Pr(y = 1|x) \approx P_{A,B}(f) \equiv \frac{1}{1 + exp(Af + B)}. \qquad (10)$$

A study comparing different methods for estimating confidences from the output of SVMs indicated this approach as the most stable [29]. The value of f in equation(10) is the decision margin obtained for the input x at time t. The A and B values are parameters that should be estimated using a set of well annotated decision margins. Here we use the Platt implementation, proposed in [31], to estimate the values of A and B.

We expect that the obtained probabilities will have values close to 1 when a frame should be classified using the selected class (hard acceptance) and close to 0 for a hard rejection. We therefore replace the output margins with the conditional probabilites obtained via equation (10). We denote them in the following as M. On the basis of the conditional probabilities $M_{ni=1}^{iC}$, with C= number of classes, for each frame n, we define the two following conditions for detecting challenging frames:

1) $M_n^i < M_{max_{i=1}^{C}}$: for each of the possible classes C none obtains a high level of confidence;

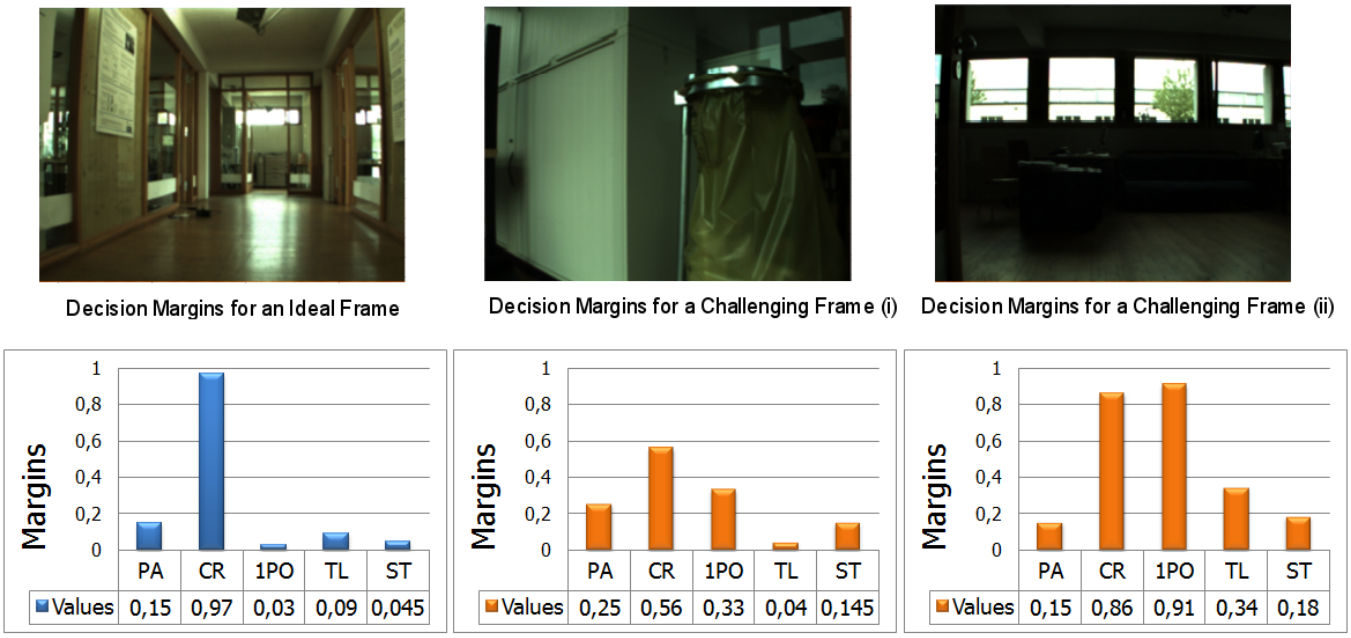

Fig. 1. Decision margins obtained for an ideal frame and two types of challenging frames.

2) $|M_n^i - M_n^j| < \Delta_{i=1}^C$: there are at least two classes with high level of confidence, but their difference is too small to allow for a confident decision.

In order to show why these two conditions are important, Figure 1 shows three examples of the conditional probabilities obtained for a frame correctly classified with high confidence (left) and two frames labelled as challenging frames (centre and right). Figure 1, centre, show an example of frame classified as challenging because of a low level of confidence (condition (1)); Figure 1, right, shows instead an example of challenging frame where there are two high and very close levels of confidence (condition (2)).

Once a frame has been identified as challenging, it is added to a set of challenging frames that will be used to retrain the classifier or discarded when the robot cross a new door. This decision will depend on the conditional probabilities for all classes since the robot crossed the last perceived door, as it will be discussed in the next section.

The value of M_{max} and Δ was selected after several preliminary results, taking into account the potential risk of retraining the classifier with a frame incorrectly classified. Using a conservative approach, a test frame p is not labelled as challenging only if the confidence value M_p^j was higher than 0.95 and also higher than $0.8 * \sum_{i=1}^C M_n^i$.

B. Detecting New Rooms

Once a frame has been classified as challenging, we might be facing two very different situations: (a) the robot has entered a room never seen before, or (b) the robot has entered a room previously seen under some unusual imaging conditions. In this Section we discuss how to detect the first case.

When a robot enters a room not seen during training, we would expect that most of the conditional probabilities for all known classes should be close to 0. Furthermore, we would expect that by looking at all frames since crossing the last detected door one would not be able to detect a dominant class label. Our proposal is to use these two behaviors as indicators of being in a new room, to do so, we have developed the following quantitative approach: starting from the moment when the robot detects crossing a door, we consider n frames and their associated conditional probabilities. If the majority of the frames are classified as challenging, then the robot has entered a new room. Quantitatively this can be measured as follows: We define the quantity $S_{i=1}^C = \sum M_{ni=1}^{iC}$, with C number of classes.

To detect a new room, at least one of the two following conditions needs to be met:

- $S_i < n * T1$: for each of the possible classes C none obtains a high level of confidence;
- $\sum_{i=1}^C S_i < n * T2$: the sum of all conditional probabilities over all frames is below n.

$T1$ and $T2$ are threshold values determined experimentally; in this paper the selected values where $T1 = 0.3$ and $T2 = 0.4$. If at least one of the two conditions is satisfied, we assume that the robot has visited a new room. In this situation, the robot will use all challenging frames (that should be similar to n) to retrain the classifier using a new label. That label can be directly generated by the robot, or the robot can ask for a new label to a human supervisor.

C. Detecting Old Rooms

After detecting a door crossing situation and having computed S_i for all classes, we can have two cases:

- The robot has entered a new room
- The robot has entered a known room.

The first case has been discussed in the last section. Here we focus on the second. Detecting challenging frames when

entering a known room is most likely related to some substantial changes in the imaging conditions, such as varying illuminations of furniture relocated. These frames, if correctly labeled, would be very valuable for the algorithm because they might contribute to avoid misclassifications in the future. Again, we propose to detect known rooms by studying the behavior of S_i. We say that the robot has entered the room class C_j if the two following conditions are satisfied:

- $S_j > T3 * n$
- $S_j > T4 * \sum_{i=1}^{C} S_i$

This situation is represented in Figure 1 left. The $T3$ and $T4$ values were experimentally selected to $T3 = 0.5$ and $T4 = 0.65$. After detecting a known room, all challenging frames are used to retraining the classifier, using C_j as the correct class. If the system is not able to assign the challenging frames to a new room, or to a known room, the frames are discarded. The whole process is illustrated in Figure 2

V. Door detection algorithm

Our system is based on the ability of the robot to detect door crossing. Current indoor robots are usually equipped with a large number of sensors, mainly visual cameras and distance sensors. Door detection has been deeply studied in literature and we can implement for instance the algorithm presented in [32].

Not all databases available as benchmarks for robot localization provide laser data, so we decided to use only the visual information for doors detection. This section illustrates a simple door detector developed for the Robot Vision challenge inside the ImageCLEF [1] campaign.

Door crossing within CLEF training sequences can be observed as two vertical rectangles with the same colour that increase their side and suddently the dissappear. We will detect that situation by extracting these rectangles and studing their width evolution when new frames are acquired. The image processing starts with a Canny filter [33] to extract all the edges of the images. After this preliminary step, we apply the Hough transform [34] for lines detection discarding all the non vertical lines. The last step to extract the rectangles is to measure the average colour value between each two vertical lines, removing non homogeneous colour distributions (blobs). Fig. 3 shows this process, where three colour homogeneous blobs are detected and two could be used to detect the door crossing.

After extracting all the key blobs from a frame, we have to study the time correspondence for these blobs between this frame and the last frames. If two blobs with the same average colour are increasing for new frames we are reaching a door and both blobs are marked as candidates. Preliminary candidates will be selected as definitive ones if one of the two blobs starts decreasing after reaching the largest size at the left (right) of the image. Figure 4 shows four consecutive training frames, where white rectangles represent blobs, preliminary candidates are labelled with a P and definitive candidates with a D. Green rectangles for the bottom images represent the time correspondence for each blob in the last frames. This idea can be extended to develope simple and efficient door crossing detectors for corridor environments, and was used to detect the door crossing in this paper.

[1] http://www.robotvision.info/

VI. Experimental Setup

In this section we describe the experimental setup used to validate our approach. Section VI-A describes the data used, and section VI-B the feature descriptors. The description of each experiment with the corresponding result is given in section VII.

A. The Database

For our experiments we used two different databases that we describe in the rest of the section.

The COLD Database The COLD database [5] contains three separate sub-datasets, acquired at three different indoor labs, located in three different European cities: the Visual Cognitive Systems Laboratory at the University of Ljubljana, Slovenia; the Autonomous Intelligent System Laboratory at the University of Freiburg, Germany; and the Language Technology Laboratory at the German Research Center for Artificial Intelligence in Saarbrücken, Germany. For each lab, image sequences of several rooms are provided, all acquired with the same camera settings.

Here we used the sub-dataset acquired in the Autonomous Intelligent System Laboratory at the University of Freiburg, Germany (COLD-Freiburg): it consists of three sets of sequences, both acquired under varying illumination conditions. Of these three sets, we chose the following two: In the first set, the robot travels across five rooms: corridor, 2-person office, printer area, bathroom and stairs area. In the second set, the robot travels across the rooms of the first set, plus other four rooms: a 1-person office, a printer area, a kitchen and a large office. Figure 5 shows some exemplar views from the second set of sequences. Each of the sequences described above were acquired under three different illumination conditions -sunny, cloudy and night. Three sequences were acquired, one after the other, for each weather condition, for a total of nine data sequences for each set.

The ImageCLEF 2010 Database The image sequences used for the Robot Vision Task at the ImageCLEF 2010 challenge evaluation were taken from the previously unreleased COLD-Stockholm database [35]. The sequences were acquired using a MobileRobots PowerBot robot platform equipped with a stereo camera system consisting of two Prosilica GC1380C cameras. The acquisition was performed on three different floors of an office environment, consisting of 36 areas (usually corresponding to separate rooms) belonging to 12 different semantic and functional categories. 8 of these semantic categories are shown in Fig.5

The robot was manually driven through the environment while continuously acquiring images at a rate of 5fps. Each data sample was then labeled as belonging to one of the areas

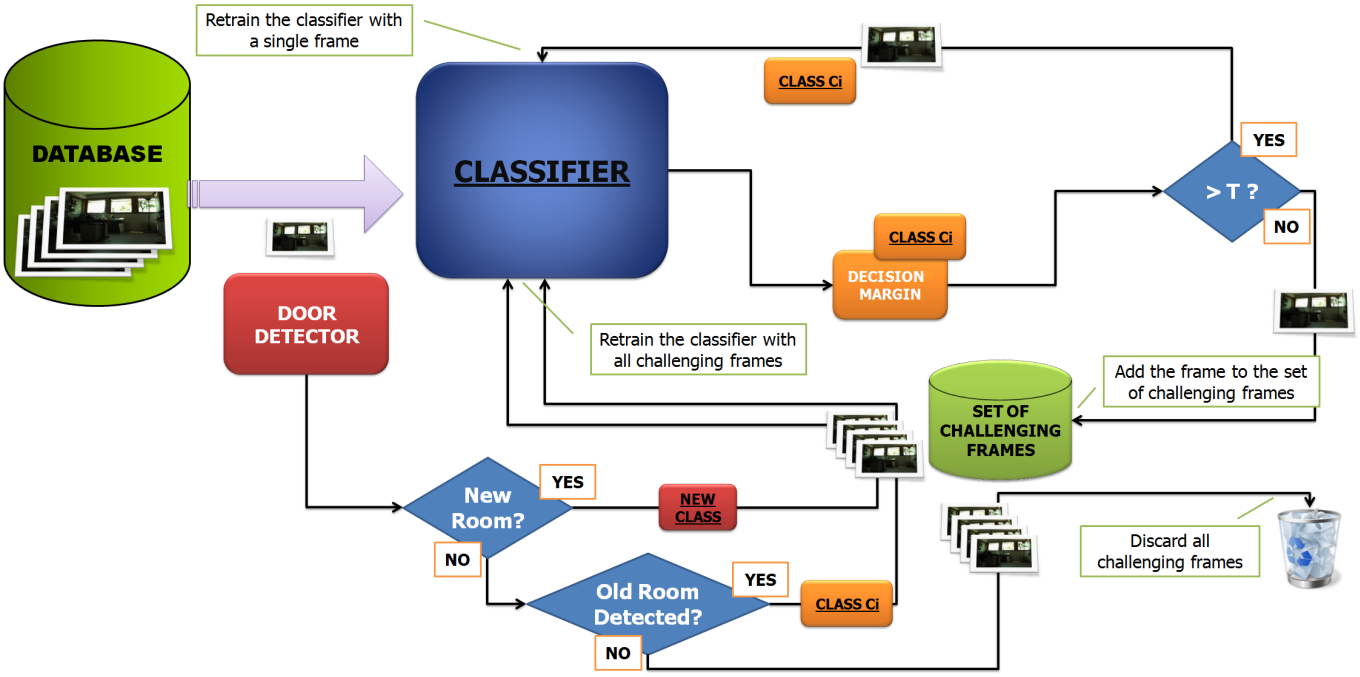

Fig. 2. Complete classification process performed by our proposal, where frames classified that obtained promising decision values (not challenging frames) are used to retrain the classifier and the set of challenging frames is processed after crossing a door.

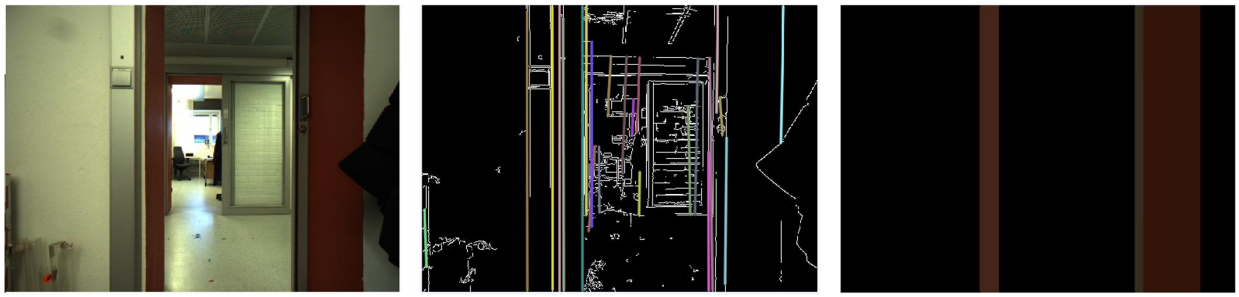

Fig. 3. Complete process to extract blobs. Left: original image. Centre: Vertical lines detection. Right: Homogeneous colour distributions between two vertical lines (BLOB extraction)

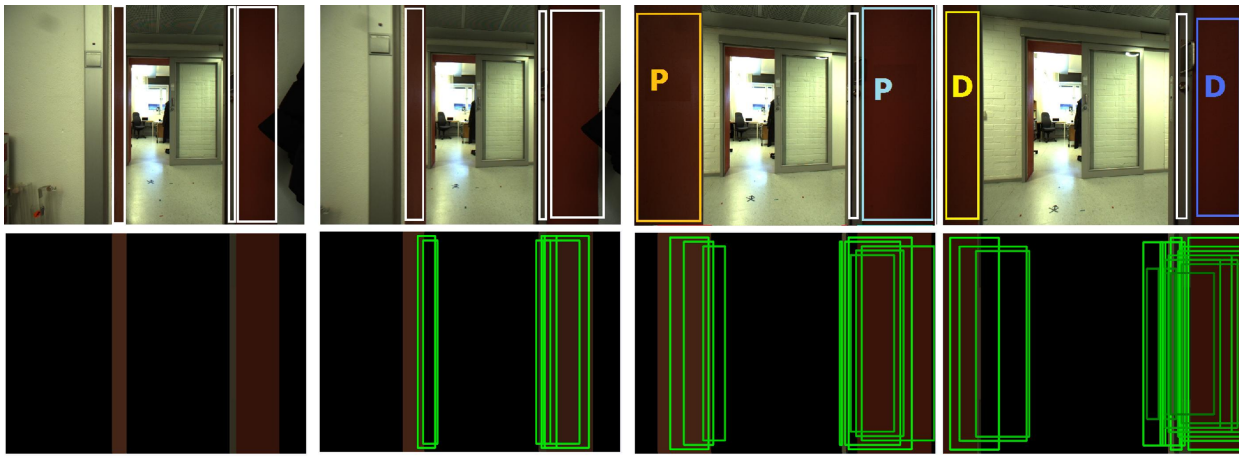

Fig. 4. Door detection for four consecutive frames. Top images are the original frames using P for preliminary candidates and D for definitive ones. Bottom images show the blobs extracted and time correspondence between them

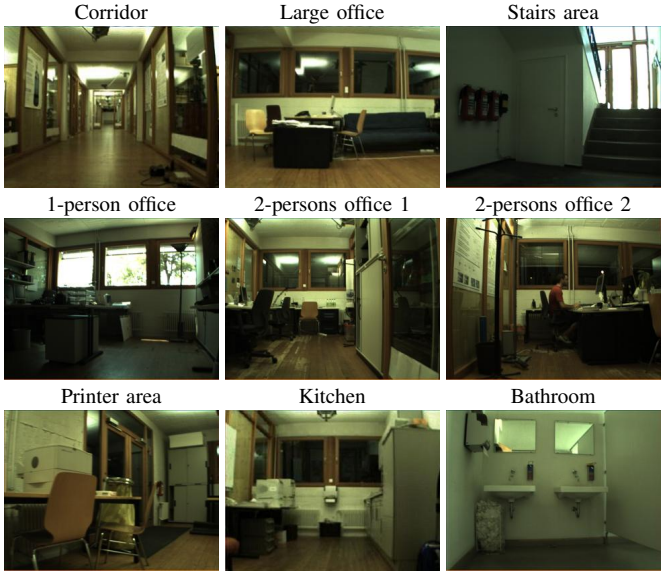

Fig. 5. Examples of images from the COLD-Freiburg database.

Fig. 6. Examples of images from the CLEF 2010 database.

according to the position of the robot during acquisition (rather than contents of the images).

Three sequences were selected for the contest: a training sequence, a sequence that should be used for validation and a sequence for testing:

- training sequence: Sequence acquired in 11 areas, on the 6th floor of the office building, during the day, under cloudy weather. The robot was driven through the environment following a similar path as for the test and validation sequences and the environment was observed from many different viewpoints (the robot was positioned at multiple points and performed 360 degree turns).
- validation sequence: Sequence acquired in 11 areas, on the 5th floor of the office building, during the day, under cloudy weather. Similar path was followed as for the training sequence; however without making the 360 degree turns.
- testing sequence - Sequence acquired in 14 areas, on the 7th floor of the office building, during the day, under cloudy weather. The robot followed similar path as in case of the validation sequence.

B. The Features

As features, we chose a variety of global descriptors representing different features of the images. We opted for histogram-based global features, mostly in the spatial-pyramid scheme introduced in [36].

This representation scheme was chosen because it combines the structural and statistical approaches: it takes into account the spatial distribution of features over an image, while the local distribution is in turn estimated by mean of histograms; moreover it has proven to be more versatile and to achieve higher accuracies in our experiments.

The descriptors we have opted to extract belong to five different families: Pyramid Histogram of Orientated Gradients (PHOG) [37], Sift-based Pyramid Histogram Of visual Words (PHOW) [38], Pyramid histogram of Local Binary Patterns (PLBP) [39], Self-Similarity-based PHOW (SS-PHOW) [40], and Compose Receptive Field Histogram (CRFH) [41]. Among all these descriptors, CRFH is the only one which is not computed pyramidly. For the remaining families we have extracted an image descriptor for every value of $L = \{0, 1, 2, 3\}$, so that the total number of descriptors extracted per image is equal to 25 ($4+4$ PHOG, $4+4$ PHOW, 4 PLBP, 4 SS-PHOW, 1 CRFH). In order to select the best visual cues to be combined together we performed a pre-selection step, namely we run some preliminary experiments to decide which combination of features was more effective. This eventually made us settle on two descriptors, PHOG L0 and Oriented PHOG L2. Their exact settings are summarized in Table I for the experiments done on the COLD database, and in Table II for the experiments done on the ImageCLEF database. These features are concatenated to generate a single feature that will be used as input for the classifier.

Descriptor	Settings	L
PHOG_{180}	range= [0, 180] and $K = 20$	$\{0\}$
PHOG_{360}	range= [0, 360] and $K = 40$	$\{2\}$

TABLE I
SETTINGS OF THE IMAGE DESCRIPTORS USED FOR THE COLD DATABASE

Descriptor	Settings
PHOG_{180}	range=[0, 180], K=20, L=$\{0, 1, 2, 3\}$
PHOG_{360}	range=[0, 360], K=40, L=$\{0, 1, 2, 3\}$
$\text{PHOW}_{\text{gray}}$	$[M, V]$=[10, 300], $r = \{4, 8, 12, 16\}$, L=$\{0, 1, 2, 3\}$
$\text{PHOW}_{\text{color}}$	$[M, V]$=[10, 300], $r = \{4, 8, 12, 16\}$, L=$\{0, 1, 2, 3\}$
$\text{PLBP}_{8,1}^{riu2}$	$[P, R]$=[8,1], RotationInvariantUniform2 version, L=$\{0, 1, 2, 3\}$
SS-PHOW	$[M, V, S, R, nRad, nTheta]$=[5,300,5,40,4,20], L=$\{0, 1, 2, 3\}$
CRFH	Gaussian-Derivatives=$\{L_x, L_y\}$, K=14, s=$\{1, 2, 4, 8\}$

TABLE II
SETTINGS OF THE IMAGE DESCRIPTORS USED FOR THE IMAGECLEF DATABASE.

VII. RESULTS

The main objective of all experiments performed is to test the feasiblity of using our system for real-case scenarios. to this end, we need to demonstrate experimentally two points: (1) that the performance of the Memory controlled OI-SVM is similar to that of the original method, and (2) that when retraining the system with new self-labeled frames, the performance of the algorithm increases.

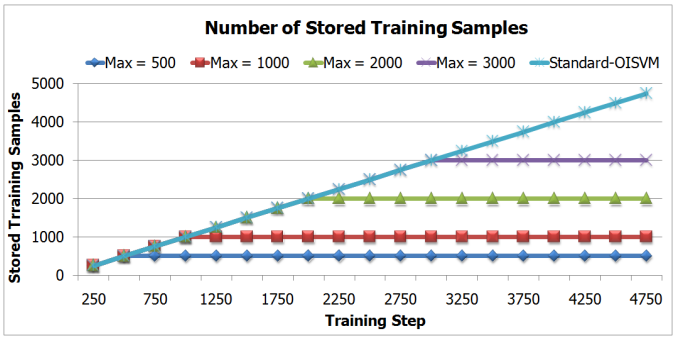

Fig. 8. Number of stored training samples for different $MaxTSs$ values

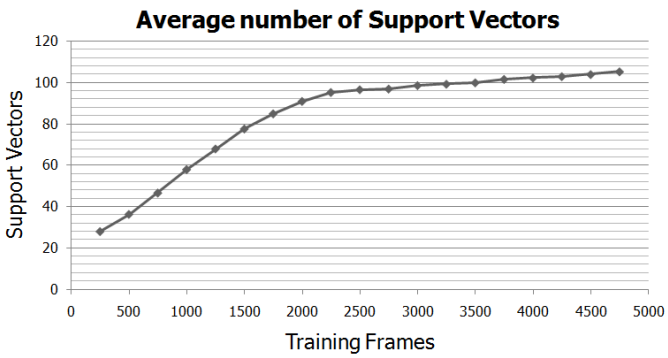

Fig. 10. Average number of support vectors stored for Experiment 1

All the experiments were performed with an SVM classifier using the χ^2 kernel, with $C = 1$, $\gamma = 1$ and $\eta = 0.25$. The ImageCLEF and the COLD-Freiburg were the databases used for the experiments. The sequences for training and testing were selected as follows:

- ImageCLEF: We just had a single combination for training/testing: the proposed training sequence was used for training and the obtained classifier was tested with the validation sequence proposed for the task and correctly labelled (the ImageCLEF test sequence is not labelled).
- COLD-Freiburg: Training always consisted of three sequences (from those proposed for training in database, left column in Figure 7), acquired one after the other, with the same illumination conditions. Testing consisted of one sequence (from those proposed for testing, right column in Figure 7), taken from those not used for training. The COLD-Freiburg database consists of sequences of images acquired with three lighting conditions (cloudy, night and sunny), and so we used 9 training/testing combinations.

A. Experiment 1: Memory-Controlled OISVM

In a first set of experiments we compared the performance of the Memory Controlled OI-SVM and the original method. We determined a priori several values for the maximum number of stored training samples $MaxTSs$ and we measured the classification rate obtained for the selected training sequence and testing sequence combination.

To compare the performance of MC-OI-SVM with that of OI-SVM, we used the 9 sequence combinations from the COLD-Freiburg. Because of the different illumination conditions (cloudy, sunny, night) there are 9 different combinations of training and test data. We performed experiments on all of them, chosing for MC-OI-SVM four different values of $MaxTSs$: (500,1000,2000,3000).

In order to show how this threshold affects the memory requirements of the system, Fig 8 shows the number of training samples stored by the algorithm common for all different $MaxTSs$ values.

Experiments were performed as follows: the classifier was incrementally trained using the training sequence and evaluated periodically (250 frames). These evaluations were performed by classifing the whole testing sequence with the classifier obtained in that time. Figure 9 presents these results, where the 9 combinations are shown: columns represents the lighting conditions used for testing and rows for training. The $x - axis$ represents the number of training samples used to generate the classifier in that point and the $y - axis$ the error rate.

It can be observed how the error rate is not affected when using $MaxTSs$ values greater than 2000, because the performance of the two algorithms is essentially the same. From this Figure, and from Figure 8, we can make two remarks: (1) with MC-OI-SVM it is possible to obtain an impressive reduction on the memory requirements of the original method with a very negligible decrease in accuracy; (2) when the $MaxTSs$ value is too close to a certain limit L_{MaxTs} the approximation affects the optimality of the solution, and the error rate starts to increase.

The optimal value of this limit L_{MaxTs} depends of the number of support vectors stored by the optimal solution (without removing any training frame). In our experiments, we also stored the number of support vectors of the original OISVM, and the average number of stored support vectors for each number of training frames is shown in Figure 10.

From this Figure and Figure 9 we can observe that problems with $L_{MaxTs} = 500$ started when the number of training frames was higher 750 and became effective for most of the combinations when the number of training frames was 1000. For $L_{MaxTs} = 1000$, problems started (if they happened) when the number of training frames was 1500. If we translate this number of training frames into support vectors (using Fig.10), problems presented for $L_{MaxTs} = 500$ when having a number of support vectors higher than 46 and for $L_{MaxTs} = 1000$ when that number was higher than 77.

It should be studied the relationship between the number of stored support vectors and the minimum value of L_{MaxTs} without decrease in accuracy. This relationship could be used for a dynamic establishment of L_{MaxTs} using the number of stored support vectors.

B. Experiment 2: Retraining the classifier with the MC-OI-SVM

In a second set of experiments, we studied the impact of retraining the system using testing frames. For this experiment, our MC-OISVM classifier included all the processing shown in Fig. 2. After classifying a new frame, such test frame can be used to retrain the classifier or added to a set of

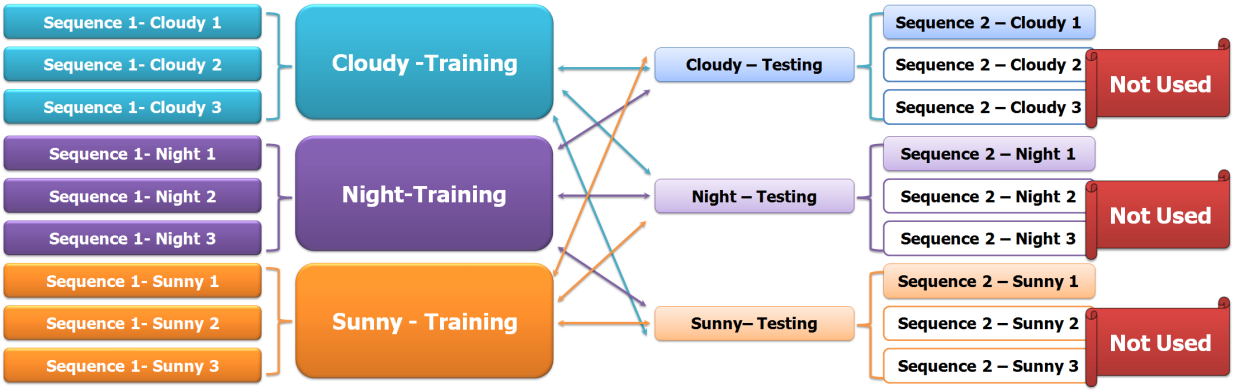

Fig. 7. 9 combinations of training and testing sequences selected from COLD-Freiburg database for the experiments 1 and 2

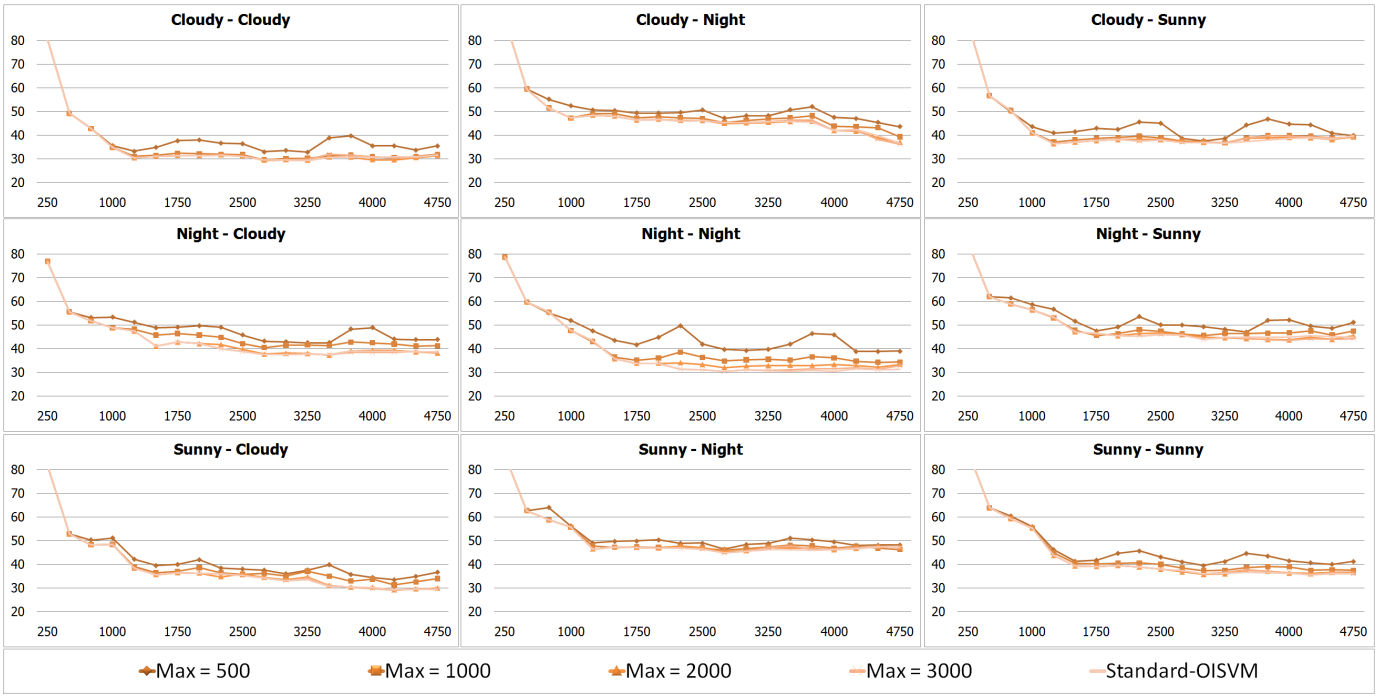

Fig. 9. Error rate (processing the whole test sequence) obtained incrementally for the 9 combinations of training and testing sequences from COLD-Freiburg database, obtained for different $MaxTSs$ values

challenging frames. The set of challenging frames can also be used to retrain the classifier, after detecting a door crossing. All settings used for the algorithm (and mentioned in Section IV-A, IV-B and IV-C) were common for all experiments. The value for L_{MaxTs} was 3000 for all experiments.

We used the same combinations and sequences for training and testing as for Experiment 1. For this experiment, instead of measuring the error rate over the complete test sequence while the classifier was being generated, we measured the relative accumulated error (RAE) while the test sequence was processed.

1) COLD-Freiburg Database: Fig.11 shows the results obtained for the 9 combinations of the COLD-Freiburg database, where each row was used for a different testing sequence and new rooms are marked using dark areas.

Adding MC-OISVM always obtained a smaller error rate than the original one, with an average improvement of 2.56% over all combinations. It can be observed how the relative error always increased for new rooms, regardless of the method used. This is because new rooms could only be detected after leaving them, so all their frames were incorrectly classified.

2) CLEF Database: The same experiment was performed using the CLEF database, with a single combination of training/testing. The obtained results can be observed in Fig. 12

The improvement obtained with the CLEF database (9.56%) was notoriously higher than those obtained with COLD-Freiburg. These results were promising for us, due to our proposal had a better behaviour facing a more challenging database (CLEF training/testing sequences were acquired on different floors).

Retraining our classifier with non-challenging frames improved the adaptability of our system to new lighting conditions. Expected lighting changes were exposed in an extreme way for our experiment, where the system was trained with

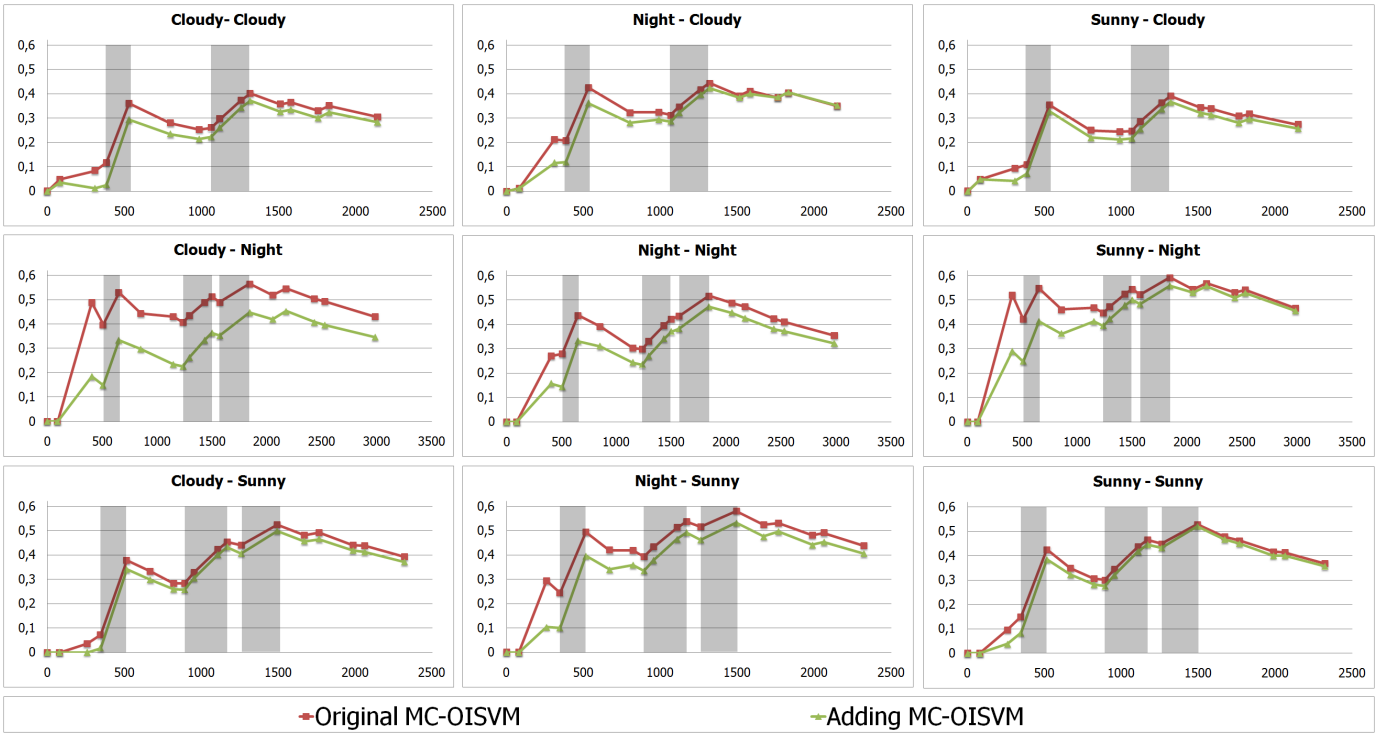

Fig. 11. Relative accumulated error obtained for the 9 combinations of training and testing sequences from COLD-Freiburg database, obtained with adding and original MC-OISVM. Dark areas represent new rooms not imaged previously

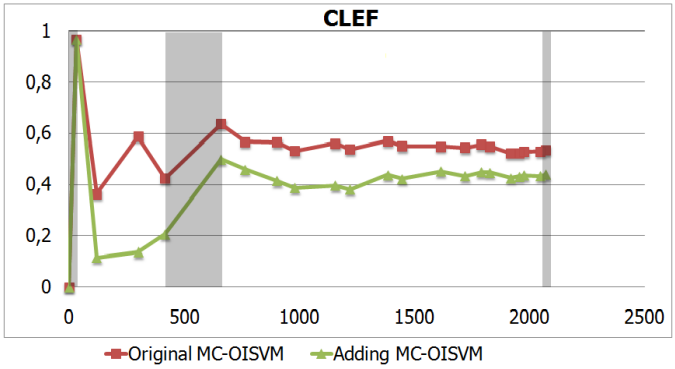

Fig. 12. Relative accumulated error obtained for CLEF database, obtained with adding and original MC-OISVM. Dark areas represent new rooms not imaged previously

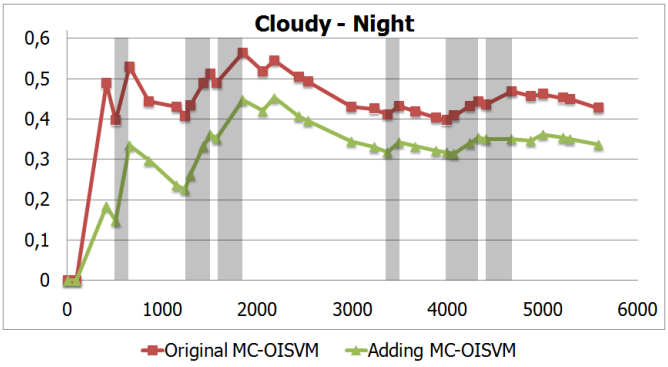

Fig. 13. Relative accumulated error for a Cloudy-Night combination

conditions totally different as those used for testing, without small or progressive variations.

C. Experiment 3: Impact of new room recognition for future recognition

The third set of experiments consisted of a deeper study on the capability of our system to recognize new rooms, and on its impact on performance over time. We performed the same experiments as in Experiment 2 but using only a new testing sequence generated by joining two of the proposed testing sequences obtained under the same lighting conditions (night). Because of that, new room detection it is supposed to improve considerably the error rate for future classifications.

After leaving a room, the set of challenging frames is studied and we determine the type of room we have visited: new room, known room or it is not possible to say the type of room without uncertainty. After a new room detection, the robot will ask for human supervision and a new label will be generated for the new spatial category. The complete set of challenging frames will be used to retrain the classifier using the new label as a class.

The experiment was performed with the same parameters as in Experiment 2, and the new testing sequence was processed after generating the classifier with the three training sequences used for the experiments 1 and 2: cloudy, night and sunny (exposed in Fig.7 left).

Figures 13, 14 and 15 present the results obtained for these combinations of training/testing sequences, where we measured the relative accumulated error.

The average improvement for the error rate using our approach over the original MC-OISVM was 8.10%. It can

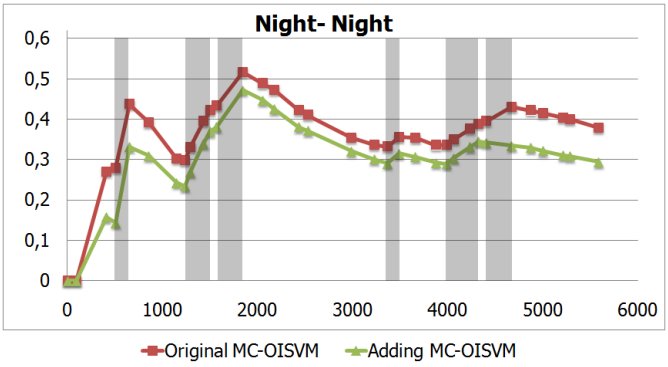

Fig. 14. Relative accumulated error for a Night-Night combination

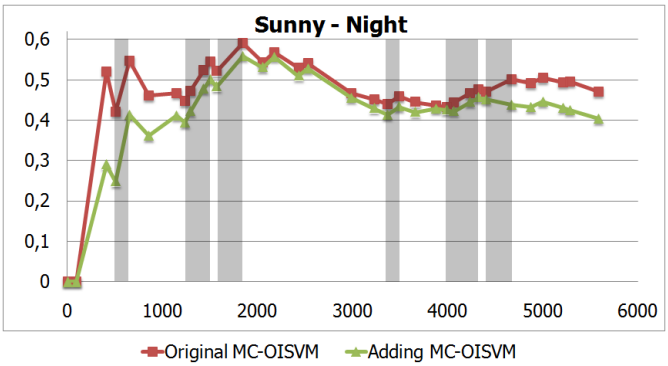

Fig. 15. Relative accumulated error for a Sunny-Night combination

be observed how the new room recognition increased the tolerance of the system to unknown rooms.

It should be pointed out the behaviour of the system with the third unknown room presented in the testing sequence: the kitchen. That room is represented (in Fig. 13, 14 and 15) by dark rectangles located between frames 1569-1841 and 4402-4672. First kitchen rectangle (third in figures) represents the first time the room appears, where all frames were misclassified for adding and original MC-OISVM. The second time this room appears (frames 4402-4672), all frames were incorrectly classified by the original MC-OISVM (the error rate increased notoriously) but perfectly labelled by the extra layer (the accumulated error rate decreased). The other two unknown rooms presented problems due to the similarity between these rooms and other previously trained: Large Office and 1 Persons Office were similar to 2 Persons Office. This similarity affected to the detection of the new room and also presented problems for future classifications.

VIII. Conclusions and Future Work

In this paper we presented an algorithm for online learning of semantic spatial concepts with a bounded memory growth, able to measure its own level of confidence when classifying incoming frames, and therefore able to decide when to ask for human annotation and when to trust its own decisions. Experiments on a subset of the challenging COLD database [5] show that our approach is able to minimize the false positives when classifying known frames, and it is able to detect new rooms, not seen during training.

This work can be continued in many ways. With respect to the confidence estimate, here we used the conditional probabilities of the SVM-based classifiers, but more elegant and sophisticated options should be explored here. Also, here we applied the method to only visual features, but this framework should work, and benefit from, multi-modal data such as laser range features. Future work will proceed in these directions.

References

[1] G. Lakoff, *Women, fire and dangerous things: what categories reveal about the mind*. The University of Chicago Press, 1990.

[2] F. Orabona, C. Castellini, B. Caputo, J. Luo, and G. Sandini, "Indoor place recognition using online independent support vector machines," in *Proc. BMVC*, vol. 7. Citeseer.

[3] M. Ullah, F. Orabona, B. Caputo, I. IRISA, and F. Rennes, "You live, you learn, you forget: Continuous learning of visual places with a forgetting mechanism," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009*, 2009, pp. 3154–3161.

[4] A. Pronobis and B. Caputo, "Confidence-based cue integration for visual place recognition," *Proc. IROS07*.

[5] ——, "COLD: COsy Localization Database," *The International Journal of Robotics Research (IJRR)*, vol. 28, no. 5, May 2009. [Online]. Available: http://www.csc.kth.se/ pronobis/research/pronobis09ijrr-cold/pronobis09ijrr-cold.pdf

[6] C. Siagian and L. Itti, "Biologically-inspired robotics vision monte-carlo localization in the outdoor environment," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'07)*, San Diego, CA, USA, October 2007.

[7] A. Pronobis, O. Martínez Mozos, and B. Caputo, "SVM-based discriminative accumulation scheme for place recognition," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, CA, USA, May 2008.

[8] A. C. Murillo, J. Kosecka, J. J. Guerrero, and C. Sagues, "Visual door detection integrating appearance and shape cues," *Robotics and Autonomous Systems*, vol. 56(6), pp. pp. 512–521, June 2008.

[9] C. Valgren and A. J. Lilienthal, "SIFT, SURF and seasons: Long-term outdoor localization using local features," in *Proceedings of the European Conference on Mobile Robots (ECMR'07)*, 2007.

[10] S. Thrun and T. Mitchell, "Lifelong robot learning," *Robotics and Autonomous Systems 15*, 1995.

[11] T. Mitchell, "The discipline of machine learning," CMU, Tech. Rep. CMU-ML-06-108, 2006.

[12] J. Malak, R.J. and P. K. Khosla, "A framework for the adaptive transfer of robot skill knowledge using reinforcement learning agents," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'01)*, 2001.

[13] G. Konidaris and A. G. Barto, "Autonomous shaping: knowledge transfer in reinforcement learning." in *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[14] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Knowledge Discovery and Data Mining*, vol. 2, no. 2, 1998.

[15] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods)*. CUP, 2000.

[16] T. Downs, K. E. Gates, and A. Masters, "Exact simplification of support vectors solutions," *Journal of Machine Learning Research*, vol. 2, pp. 293–297, 2001.

[17] Y. Engel, S. Mannor, and R. Meir, "Sparse online greedy support vector regression," in *Proceedings ECML'02*, 2002.

[18] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Proceedings of NIPS'00*, 2000, pp. 409–415.

[19] F. R. Bach and M. I. Jordan, "Predictive low-rank decomposition for kernel methods," in *Proceedings of ICML'05*, 2005.

[20] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, 2004.

[21] S. S. Keerthi and D. DeCoste, "A modified finite newton method for fast solution of large scale linear SVMs," *Journal of Machine Learning Research*, vol. 6, pp. 341–361, 2005.

[22] S. S. Keerthi, O. Chapelle, and D. DeCoste, "Building support vector machines with reduced classifier complexity," *Journal of Machine Learning Research*, vol. 8, pp. 1–22, 2006.

[23] K. Crammer, J. Kandola, and Y. Singer, "Online classification on a budget," in *Advances in Neural Information Processing Systems 16*, 2003.

[24] J. Weston, A. Bordes, and L. Bottou, "Online (and offline) on an even tighter budget," in *Proceedings of AISTATS 2005*, R. G. Cowell and Z. Ghahramani, Eds., 2005, pp. 413–420.

[25] J. Kivinen, A. Smola, and R. Williamson, "Online learning with kernels," *IEEE Trans. on Signal Processing*, vol. 52, no. 8, pp. 2165–2176, 2004.

[26] L. Cheng, S. V. N. Vishwanathan, D. Schuurmans, S. Wang, and T. Caelli, "Implicit online learning with kernels," in *Advances in Neural Information Processing Systems 19*, 2007.

[27] O. Dekel, S. Shalev-Shwartz, and Y. Singer, "The Forgetron: A kernel-based perceptron on a budget," *SIAM Journal on Computing*, vol. 37, no. 5, pp. 1342–1372, 2007.

[28] B. C. M. M. Ullah, F. Orabona, "you live, you learn, you forget: continuous learning of visual places with a forgetting mechanism," in *Proceedings of IROS'09*, 2009.

[29] S. R
"uping, *A simple method for estimating conditional probabilities for svms*. Citeseer, 2004.

[30] J. Platt, Ô.
"OÓ, and Ó. ÓÑ, "Probabilistic outputs for support vector machines," *Bartlett P. Schoelkopf B. Schurmans D. Smola, AJ, editor, Advances in Large Margin Classifiers*, pp. 61–74.

[31] H. Lin, C. Lin, and R. Weng, "A note on Platts probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, no. 3, pp. 267–276, 2007.

[32] D. Anguelov, D. Koller, E. Parker, and S. Thrun, "Detecting and modeling doors with mobile robots," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3777–3784.

[33] J. Canny, "A computational approach to edge detection," *Readings in computer vision: issues, problems, principles, and paradigms*, vol. 184, 1987.

[34] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 12, no. 2, pp. 111–122, 1981.

[35] A. Pronobis, M. Fornoni, H. I. Christensesn, and B. Caputo, "The robot vision track at imageclef 2010," in *Working Notes of ImageCLEF 2010*, 2010.

[36] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006.

[37] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, p. 408.

[38] ——, "Image classification using random forests and ferns," in *International Conference on Computer Vision*. Citeseer, 2007, pp. 1–8.

[39] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," *Computer Vision-ECCV 2000*, pp. 404–420, 2000.

[40] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, 2007, pp. 1–8.

[41] O. Linde and T. Lindeberg, "Object recognition using composed receptive field histograms of higher dimensionality," in *Proc. ICPR*. Citeseer, 2004.