The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

You Live, You Learn, You Forget: Continuous Learning of Visual Places with a Forgetting Mechanism

Muhammad Muneeb Ullah
INRIA Rennes, IRISA
Rennes, France

Francesco Orabona, Barbara Caputo
Idiap Research Institute
Martigny, Switzerland

Abstract— **To fulfill the dream of having autonomous robots at home, there is a need for spatial representations augmented with semantic concepts. Vision has emerged recently as the key modality to recognize semantic categories like places (office, corridor, kitchen, etc). A crucial aspect of these semantic place representations is that they change over time, due to the dynamism of the world. This calls for visual algorithms able to learn from experience while at the same time managing the continuous flow of incoming data. This paper addresses these issues by presenting an SVM-based algorithm able to (a) learn continuously from experience with a fast updating rule, and (b) control the memory growth via a random forgetting mechanism while at the same time preserving an accuracy comparable to that of the batch algorithm. We apply our method to two different scenarios where learning from experience plays an important role: (1) continuous learning of visual places under dynamic changes, and (2) knowledge transfer of visual concepts across robot platforms. For both scenarios, results confirm the effectiveness of our approach.**

I. INTRODUCTION

A major requirement for having robots at home is that their space representation must at least partially overlap with our own. 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. Over the past years, impressive progresses have been done in developing methods for localization and mapping that makes it possible for an autonomous robot today to traverse and map substantial work-spaces [15], [3], [10]. Still, these representations are mostly laser-based and contain little semantic information. A growing number of research efforts points towards vision as the sensor modality able to provide the necessary information for generating augmented localization maps [11], [4].

Focusing on indoor, human-made environments, the most intuitive semantic concept that one can wish to extract from visual information is that of places, intended as rooms with different functionalities. Visual place recognition for mobile robots is a challenging task, for two main reasons: (a) when the task is to recognize a specific place, like my office, my kitchen etc, the challenge lies in the dynamic aspect of the visual appearance, due to the place being used (b) when the task is recognizing a generic place, like a kitchen or a bathroom never seen before, the challenge is being able to exploit the knowledge on models previously seen to generalize to the newly encountered place. Both problems can be tackled by algorithms able to *learn from experience*. In the first case, learning from experience makes the model able to adapt in time to changes. We refer to this as continuous learning. In the second case, learning from experience means using in a principled way models learned before that might contain useful information. We refer to this as knowledge transfer.

Ideally, such an algorithm should satisfy three main requirements, namely: *(a) Limited resources.* Artificial cognitive systems are required to perform human-like tasks in every day scenarios. The complexity and richness of the stimuli to acquire and analyze, for each sensory channel, is in general very high. Decisions must be taken keeping into account all the available information, so to react and interact with the environment actively. *(b) Optimality.* Nevertheless, the ratio of correct recognition of the essential characteristics of the environment must not be affected. Autonomous systems must guarantee an optimal performance for each sensory channel, so to minimize mistakes. *(c) Speed.* Lastly, the system must be able to adapt (training) and operate (testing) on-line, that is, quickly. It is impossible to predict a priori how a room is going to be redecorated in two years from now. As the environment around us evolves in time, an autonomous agent should do the same.

This paper presents an SVM-based algorithm able to learn incrementally visual place models from experience. We build on previous work on approximate incremental SVM [14], where the same accuracy of the batch method is achieved at the expenses of a continuous memory growth. Here we overcome this problem, by integrating in the previous approach a random forgetting mechanism that capitalizes on the new information as it becomes available. This makes the algorithm able to adapt to the changes in the environment while at the same time maintaining a low memory requirement. Furthermore, the forgetting mechanism allows to gradually expel the old knowledge that could become a possible source of misleading information. This is particularly important when the algorithm is used for knowledge transfer. The resulting algorithm satisfies the requirements of performance, speed and limited memory growth. We tested our approach on the two scenarios outlined above, namely continuous learning of visual places in dynamic environments, and knowledge transfer across place concepts. We conducted experiments on two recently introduced databases [16], [9], benchmarking our method against the algorithm of Luo et al [8]. Results show that we achieve in both scenarios an accuracy comparable to that of the batch method, while considerably reducing the memory requirements with respect to the method of Luo et al.

The rest of the paper is organized as follows: we describe our algorithm in section II. Section III describes the experimental setup adopted in the paper, and section IV reports on our experiments and results. We conclude with a summary.

II. THE ALGORITHM

The place recognition systems we consider are based on the Support Vector Machine (SVM) classifier [2]. For a classification problem, given a set of training points, SVM training involves solving a constrained quadratic optimization problem. The solution is always expressed as a linear combination of non-linear functions, *kernels*, evaluated on a subset of training points, called *support vectors*. Hence the complexity of the function at testing time is directly proportional to the number of support vectors. Note that all the data must be available beforehand and the complexity of the training is at least quadratic in the number of samples. From the above considerations it is clear that standard SVM cannot be used in a continuous incremental learning framework.

To overcome this problem the fixed-partition technique [14] has been proposed. In this method, the training data is partitioned in batches of fixed size k, $\boldsymbol{T} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_n\}$, and the system is allowed to retrain after each batch of data, generating an updated classifier at that incremental step. In particular at step i the system is retrained using as training set the union of the support vectors at step $i-1$, $\boldsymbol{SV}_{i-1}$ and the batch of data i, $\boldsymbol{T}_i$ [14]. This method comes from the property of support vectors to "summarize" data, that is, removing the vectors that are not support vectors from a data set. Re-training will result in the same solution obtained training on the entire dataset [2].

A natural way to use the fixed-partition technique in the transfer of knowledge setting is to use the support vectors of a previously trained machine as a starting point for the learning procedure. This approach has been proved to work in the Memory-Controlled method [8], showing a performance gain in the first steps of learning compared to the system that does not use any prior knowledge. However, given that the number of support vectors is proportional to the number of training samples [13], this strategy brings longer training and testing times. Even if this method controls the growth of support vectors, however the final size of the trained machine is still bigger than without transfer of knowledge. In particular, at each incremental step, the Memory-Controlled algorithm preserves all the support vectors which cannot be expressed as a linear combination of others in the feature space. This behavior leads to an accumulation of all the linearly independent support vectors during the course of incremental learning.

A. The Random-Forget Method

We propose a modification of the fixed-partition algorithm that we call Random-Forget method. The core idea is that it is possible to reduce the size of a learned classifier at each incremental step by eliminating some randomly selected support vectors, while keeping the original classification performance intact. This takes inspiration from the work of Reduced SVM [6], where just a random subset of points is used as support vectors, gaining in training and testing time, but without losing much in performance. Consequently, our algorithm will not be trapped in a situation where it is impossible to eliminate any of the stored support vectors simply because of the unavailability of linearly dependent ones among them (as it is the case for the Memory-Controlled approach). In this way, our algorithm is expected to achieve a considerable reduction in the memory requirements with a recognition performance comparable to that of the Memory-Controlled algorithm. Fig. 1 explains our approach schematically. As in the Memory-Controlled algorithm, at the first step the prior knowledge, $\boldsymbol{PK}$, in form of the support vectors of another trained system, is fed to the system as training samples. Then every time the system receives new data, it first try to randomly discard old samples.

Of course an indiscriminate discarding of the old vectors at each step would bring a decrease in performance. The optimal strategy would be to randomly discard vectors until the performance on the test set does not drop. Given that the algorithm cannot obviously access to the test data, the incoming data itself is used to test the performance of the reduced model. Hence at each time step the incoming data is used as a proxy of the test data, before being used as training data. Four different trials to reduce the training set by discarding are attempted. If in all the four attempts the reduced training set would result in a reduced performance on the new batch of data, the incremental re-training goes on normally, as in the fixed-partition method. A parameter, $\boldsymbol{N}$, controls the percentage of stored support vectors that the algorithm tries to forget at each incremental step. The pseudo-code of the algorithm is summarized in Algorithm 1.

III. EXPERIMENTAL SETUP

In this section we describe the experimental setup used for testing our algorithms in the two scenarios of interest, namely continuous learning and knowledge transfer. Section III-A describes briefly the IDOL2 database [9], used in the first scenario. Section III-B describes instead the COLD database [16], used in the knowledge transfer scenario. Section III-C describes the visual features used in our experiments.

A. The IDOL2 Database

The IDOL2 database (Image Database for rObot Localization 2) is comprised of 24 image sequences acquired using a perspective camera mounted on two mobile robot platforms. The acquisition was performed within an indoor laboratory environment consisting of five rooms: One-person Office (OO), Two-persons Office (TO), CoRridor (CR), KiTchen (KT) and Printer Area (PA). The sequences were acquired under different weather and illumination conditions and across a time span of six months. Fig. 2 presents some sample images from the database acquired by both robots

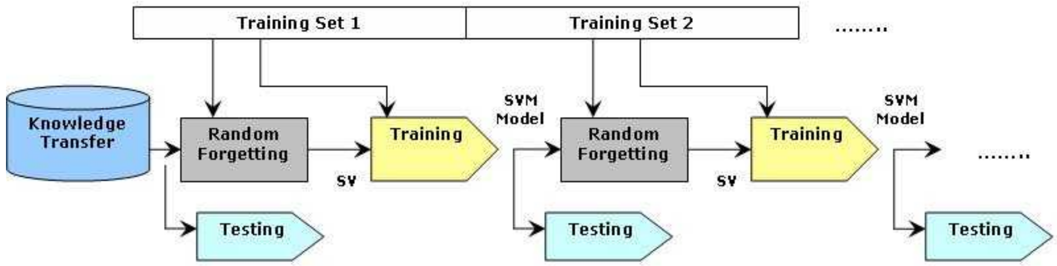


Fig. 1. A schematic flow of data in the Random-Forget approach.

Fig. 2. Pictures taken from the IDOL2 database illustrating the appearance of the five rooms from the point of view of both robotic platforms.

Algorithm 1 Pseudo-code of the Random-Forget method.

Parameters: $N, \boldsymbol{PK}, TEST\_SET$
$Model$=$\boldsymbol{PK}$
Test $Model$ on $TEST\_SET$
for each incremental step $t = 1, \ldots, n$ **do**
  $\boldsymbol{SV}$ = support vectors of $Model$
  Receive data $\boldsymbol{T}_i$
  Test $Model$ on $\boldsymbol{T}_i$, CR = classification rate
  $loopCounter = 1$
  repeat
    Discard at random $N\%$ vectors from $\boldsymbol{SV}$
    Train $Model'$, $\boldsymbol{SV'}$ = support vectors of $Model'$
    Test $Model'$ on $\boldsymbol{T}_i$, CR' = classification rate
    $loopCounter = loopCounter + 1$
  until $loopCounter == 4$ OR $CR' \geq CR$
  if $CR' \geq CR$ **then**
    Train $Model$ with $\boldsymbol{SV'} \cup \boldsymbol{T}_i$
  else
    Train $Model$ with $\boldsymbol{SV} \cup \boldsymbol{T}_i$
  end if
  Test $Model$ on $TEST\_SET$
end for

from very close viewpoints, illustrating the difference in visual content. We chose this database because its dimension and structure makes it ideal for the continuous learning scenario, while at the same time making it straightforward to benchmark our results with those presented in [8].

B. The COLD Database

The COLD (COsy Localization Database) database is a new collection of image sequences consisting of three separate sub-datasets, acquired at three different indoor labs, located in three different European cities: Saarbrucken, Freiburg and Ljubljana. The same camera settings, consisting of a perspective and omni-directional cameras, mounted together on a portable socket, were used on the mobile platform available at each lab. For each lab, the acquisition was performed in several rooms with few rooms commonly found at the other two labs. The sequences were acquired under different weather and illumination conditions, and across a time span of two/three days. To evaluate our approach in the second application scenario, i.e. knowledge transfer, we selected the COLD-Freiburg and the COLD-Ljubljana sub-datasets with four rooms in common: Two-persons Office (TO), CoRridor (CR), Printer Area (PA) and Bath Room (BR). Fig. 3 shows some selected images from the COLD-Ljubljana and the COLD-Freiburg sub-datasets.

C. Image Features Representation

For image features, we used Harris-Laplace [5] as a detector and SIFT [7] as a descriptor. These features have repeatedly proved successful for the problems of visual place recognition [12] and object recognition [17]. The main reason for their optimal performance is their local nature, which makes them invariant to different variations and hence, capture significant features that are likely to appear again under different settings.

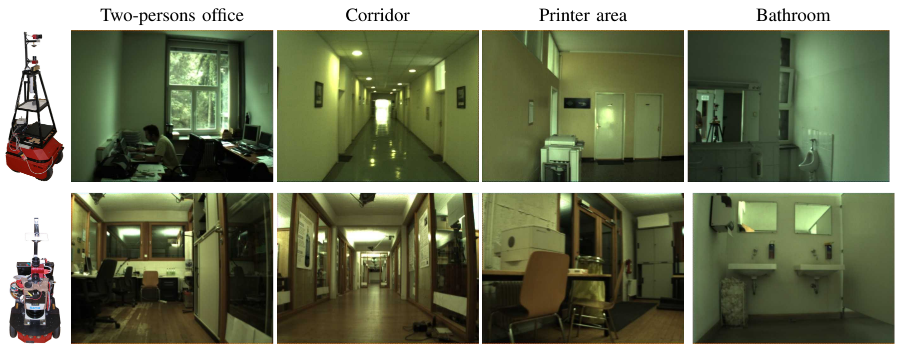


Fig. 3. Pictures taken from the COLD-Ljubljana (upper) and the COLD-Freiburg (lower) sub-datasets illustrating the appearance of four similar rooms acquired by two different platforms at two different labs.

IV. EXPERIMENTAL RESULTS

We conducted two different categories of experiments to evaluate our method. In all the experiments, the Memory-Controlled system was used with the same parameter values reported in [8]. The Random-Forget, however, was evaluated for different values of its threshold parameter. The evaluation was performed with our extended version of the *libsvm* library [1], with SVM and kernel parameters determined via cross validation. As the number of images in each sequence varied across rooms, each room was considered separately during the test experiments. The overall classification rate was then computed as an average, so that the results from each room contributed equally. Section IV-A presents the results when the systems had to learn continuously in dynamic environments, and Section IV-B reports the results when the systems had to perform knowledge transfer across place concepts.

A. Continuous Learning of Visual Places in Dynamic Environments

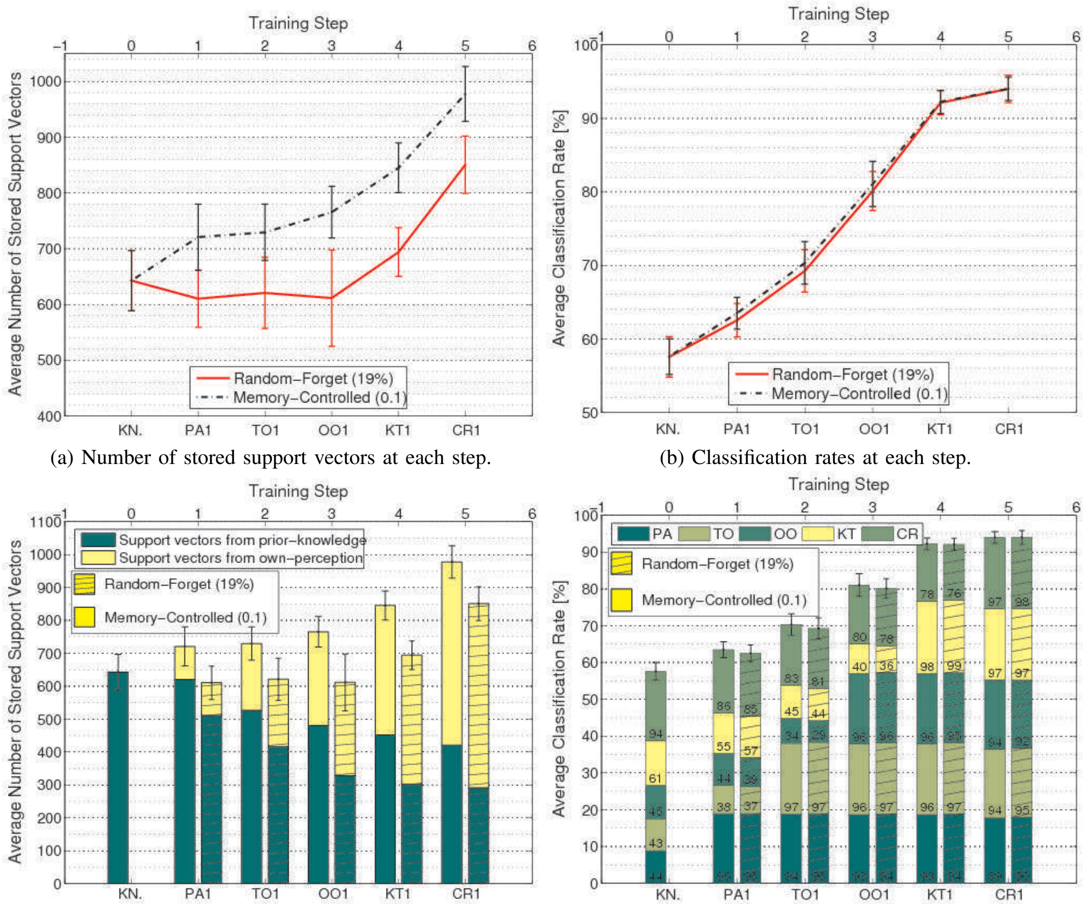
Here the objective was to study the behavior of the two algorithms, namely Random-Forget and Memory-Controlled, in a scenario where they have to learn continuously while at the same time performing recognition. For these experiments, both algorithms always employed a complete model, learned on one robot, to continue learning and recognition on the other one. This setup allowed us to evaluate the two methods against variations caused by (a) the natural variability of the environment which appears in time, and (b) the different height of the camera in the two robot platforms. As the Memory-Controlled algorithm always chooses the vectors to be discarded between those coming from the prior knowledge [8], we constrained the Random-Forget algorithm to do the same for a fair comparison. We performed two different experiments:

Room by Room Learning: In the first series of experiments, the methods were incrementally trained in a room by room (i.e. class by class) update scenario. Training was performed on one image sequence; the corresponding sequence acquired under roughly similar conditions, was used for testing. The prior-knowledge model was built on one image sequence acquired under the same illumination conditions and at close time as the training one, but using a different robot platform. As there were five classes in total, training was performed in 5 steps.

Learning from the experience of another platform implies a potentially enormous growth in the memory requirements. To evaluate this behavior in relation to its effects on performance, this experiment evaluates the two algorithms when they are trained on two sequences. We considered 6 different orderings of the sequences used as training, testing, and prior-knowledge sets with the same order of rooms (PA, TO, OO, KT, CR). Here we report average results with standard deviations. Fig. 4(a) & (b) present the average results of the two systems at each incremental step. Fig. 4(c) & (d) provide a detailed analysis of the number of stored support vectors and classification rates at each step for the two approaches.

It is evident from Fig. 4(a) & (b) that the Random-Forget achieves a great reduction in the memory requirements with approximately the same classification performance obtained by the Memory-Controlled. Fig. 4(c) shows that both methods gradually adapt to their own perception of the environment while forgetting the old knowledge. However, this phenomenon is more prominent in the Random-Forget case. As it is well known that both the training and testing time of an SVM crucially depend on the number of samples considered and the number of support vectors found as well, results illustrate that the Random-Forget method is suitable for on-line learning due to its speed.

Frames by Frames Learning: The second series of experiments wanted to analyze the behavior of the two methods when they had to perform in an on-line learning scenario. For each incremental update, we used a certain number of consecutive frames taken from the training image sequence. Again, the algorithms were incrementally trained on one sequence, and a corresponding sequence was used as a test set. The prior-knowledge model was built using two complete sequences acquired by the other platform, under the same illumination conditions and at a very close time. We considered the case when the update was performed using 30 frames per step. Thus, for each experiment, it took more than 30 incremental steps in total to complete a sequence. The experiment was repeated 6 times for different orderings of training, testing, and prior-knowledge sets. Since the number



(a) Number of stored support vectors at each step.

(b) Classification rates at each step.

(c) Comprehensive comparison of the number of stored support vectors at each step.

(d) Comprehensive comparison of the classification rates at each step.

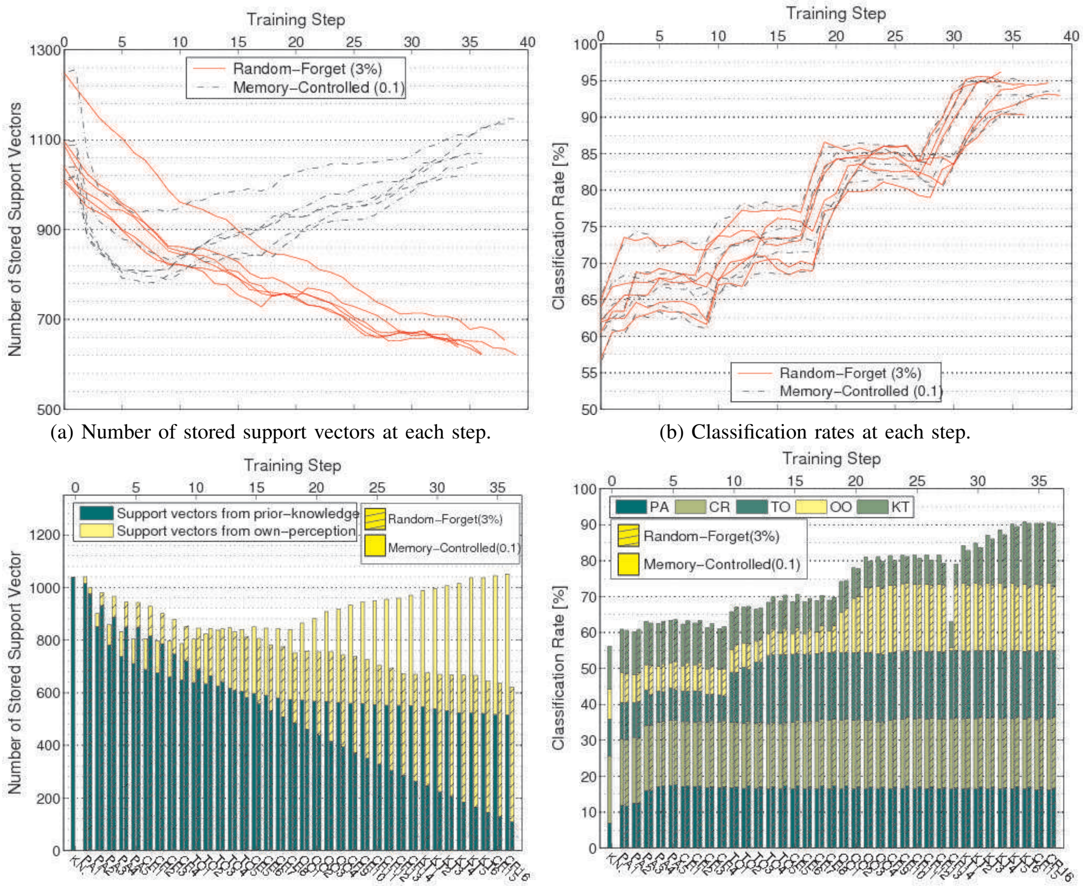
Fig. 4. Average results of the continuous learning experiments when the updation was performed room by room. Fig. 4(a) & (b) compare the total number of stored support vectors and the final classification rates for both systems. Fig. 4(c) & (d) present a comprehensive comparison: amount of support vectors in the final model that originated from the prior-knowledge and the classification rates obtained for each of the rooms. In all the plots, the first step "KN" corresponds to the results obtained for the prior-knowledge before any update was performed.

of training steps varied (due to a different number of images in each sequence), we report all the results separately. Fig. 5(a) & (b) present the total number of stored support vectors and the classification rates at each step, for all the six experiments. Fig. 5(c) & (d) report the results for one of the six experiments to allow a detailed analysis.

Fig. 5(a) clearly show that the Random-Forget method is highly suitable for on-line learning in terms of memory requirements and hence speed. With the same classification performance as that of the Memory-Controlled (Fig. 5(b)), the trend depicted in Fig. 5(a) for the Random-Forget is remarkable. At the beginning (first 10 steps of Fig. 5(a)), the Memory-Controlled indeed achieves a good reduction in the memory requirements compared to the Random-Forget but later on, it appears that the Memory-Controlled gives up in forgetting the prior-knowledge (Fig. 5(c), green portion of the bars). The Random-Forget, however, keeps on doing its job and achieves a steady reduction in the memory requirements without any loss of classification performance. We interpret this result as follows: the Memory-Controlled algorithm discards support vectors based on their linear independence [8]. Therefore, during the first steps, it discards as many support vectors as possible. Later on, when the linear dependence among the stored support vectors of the prior-knowledge is eliminated, it starts accumulating support vectors.

B. *Knowledge Transfer Across Place Concepts*

The basic idea here was to study whether the Random-Forget algorithm can be used for knowledge transfer across place concepts. This means that when a model is trained on some places at a particular lab, we would like to transfer it to another similar lab and be able to effectively generalize to places that were not seen before, in a continuous learning fashion. More specifically, a robot at a particular lab has to employ a model trained on some other robot at some other similar lab, to boost its continuous learning while performing recognition. For these experiments, we selected the COLD-Freiburg and the COLD-Ljubljana sub-datasets, which were collected by two different robots at two different labs. The two datasets have four rooms in common (TO, CR, PA, and BR). The algorithm always employed a complete model



(a) Number of stored support vectors at each step.

(b) Classification rates at each step.

(c) Detailed comparison of the number of stored support vectors at each step.

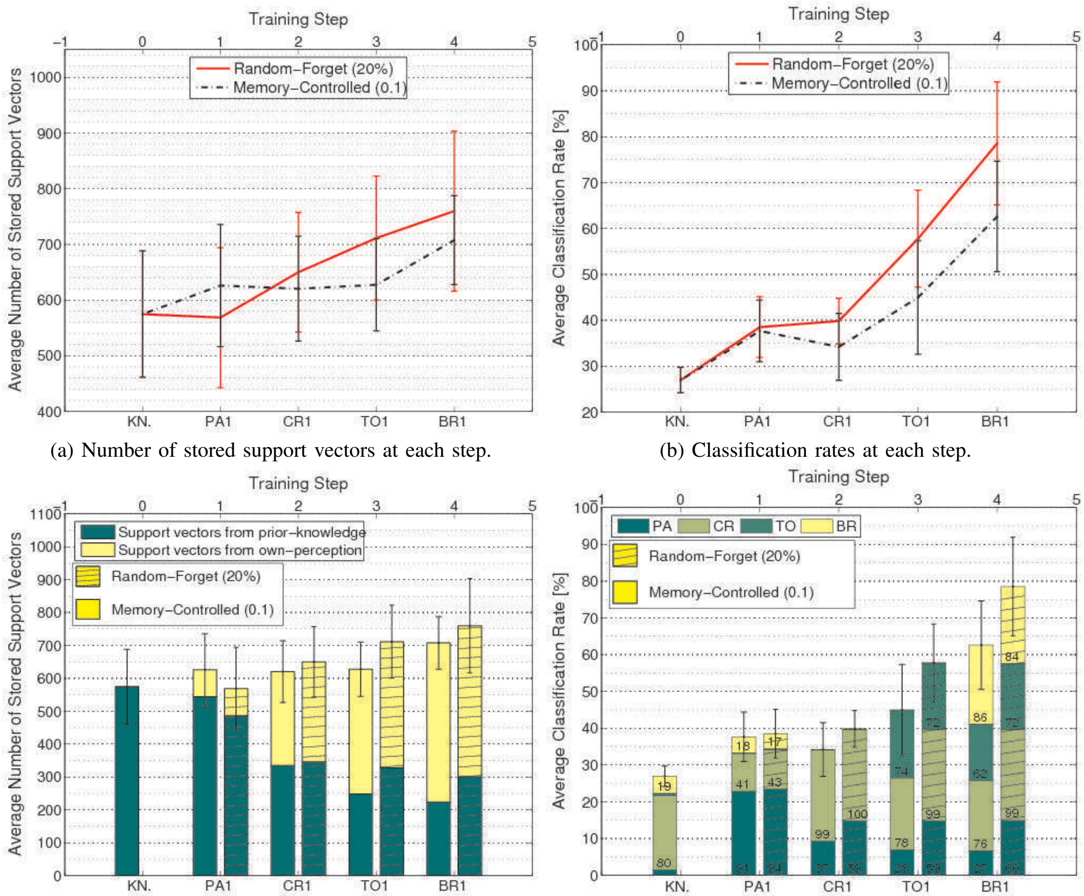
(d) Detailed comparison of the classification rates at each step.

Fig. 5. Average results of the continuous learning experiments when the updation was performed frames by frames. Fig. 5(a) & (b) compare the total number of stored support vectors and the final classification rates for all the six experiments. Fig. 5(c) & (d) present detailed results for one representative experiment: amount of support vectors in the final models that originated from the prior-knowledge and the classification rates obtained for each of the rooms. The label below each bar indicates the batch of data used for the incremental update.

learned by a different robot at a geographically distinct lab, to continue learning and recognition at its own lab. As done in the previous set of experiments, we benchmarked against the Memory-Controlled algorithm. Again, both methods performed forgetting on the prior-knowledge. We conducted similar experiments as that of the first scenario:

Room by Room Learning: The algorithms were incrementally trained room by room on one image sequence; the corresponding sequence acquired under roughly similar conditions was used for testing. The prior-knowledge model was built on one image sequence acquired under the same illumination conditions as the training one, but using a different robot platform at a different lab environment. The training was performed in 4 steps as there were four classes in total. We considered 6 different orderings of the sequences used as training, testing, and prior-knowledge sets with the same order of rooms (PA, CR, TO, BR). Here we present average results with standard deviations. Fig. 6(a) & (b) present the average results of the two systems at each incremental step. Fig. 6(c) & (d) provide a detailed analysis of the stored support vectors and classification rates at each step.

Fig. 6(a) & (b) show that Random-Forget dominates in terms of the classification accuracy, whereas, Memory-Controlled dominates in terms of the memory reduction. The results are somewhat different from the ones obtained in the first scenario. Nonetheless, an important characteristic of the Random-Forget method is revealed here. The Random-Forget will always maintain a certain level of classification accuracy while forgetting the prior-knowledge. As the classification task in this scenario is much harder than the one attacked in the previous scenario, due to the intrinsic challenge of the COLD database, the Random-Forget went for good classification accuracy while suffered in terms of memory requirements. This behavior is in accordance with the design of the method, which forgets the old knowledge based on its performance on the new incoming data. Fig. 6(d) illustrates that since the change from one model to another was abrupt (reflecting the change of geographical location), both algorithms suffered severely at the first step (PA1), and then gradually achieved a reasonable accuracy at the end. Moreover, both algorithms dropped accuracy on the previously learned classes, but Random-Forget was less



(a) Number of stored support vectors at each step.

(b) Classification rates at each step.

(c) Comprehensive comparison of the number of stored support vectors at each step.

(d) Comprehensive comparison of the classification rates at each step.

Fig. 6. Average results of the knowledge transfer experiments when the updation was performed room by room. Fig. 6(a) & (b) compare the total number of stored support vectors and the final classification rates for both systems. Fig. 6(c) & (d) present a comprehensive comparison.

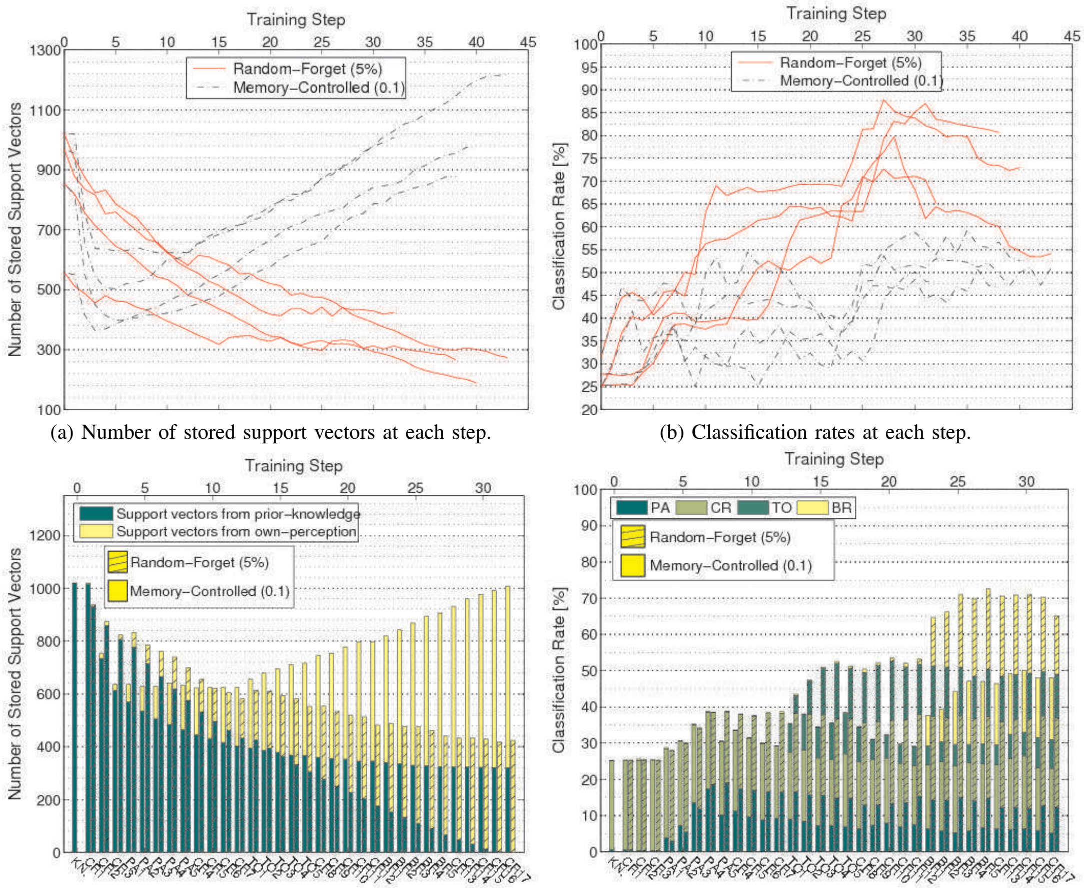
affected (see Fig. 6(d), accuracy of 'PA' at the last step).

Frames by Frames Learning: The on-line learning simulation here was formulated with 50 frames per update. This was to facilitate the execution of experiments in a reasonable number of steps, as a sequence in COLD contains roughly twice as many images as one in IDOL2. The algorithms were incrementally trained on one sequence, and a corresponding sequence was used for testing. The prior-knowledge model was built using two complete sequences acquired under the same illumination conditions by the other platform at the different lab. The experiment was repeated 4 times for different orderings of training, testing, and prior-knowledge sets. Again, the number of steps for each experiment was different, so we report the results for each experiment separately. Fig. 7(a) & (b) present for all the four experiments, the total number of stored support vectors and the classification rates at each step. This shows the general behavior of the two methods. Fig. 7(c) & (d) report the results for one of the four experiments to allow a detailed analysis.

The results in Fig. 7(a) & (b) illustrate that the Random-Forget totally outperforms the Memory-Controlled in this setup. Random-Forget achieved an enormous reduction in the memory requirements with improved classification accuracy compared to the Memory-Controlled. It is important to note in Fig. 7(a) that the growth in memory of the Memory-Controlled is almost linear, which makes it unsuitable for on-line learning. Fig. 7(d) again points out the difficulty of this problem. As similar rooms across the two labs had high within-class variability, the two methods had to learn on a whole sequence to achieve a reasonable accuracy. It is interesting, however, that when the Random-Forget was finished with the prior-knowledge (Fig. 7(c), last two steps), it still achieved better accuracy than the Memory-Controlled.

V. SUMMARY AND CONCLUSION

In this paper we presented an SVM-based algorithm able to build incrementally visual models of places like kitchens, offices, corridors and so forth. The method combines a high accuracy and fast learning rule with a bounded memory growth. This last property is achieved by a random forgetting mechanism that capitalizes on new information as it becomes available. This makes the algorithm able to adapt to the changes in the environment while at the same time storing a low number of support vectors in the visual place models. Moreover, thanks to the forgetting mechanism, the method is able to gradually expel the old knowledge that could



(a) Number of stored support vectors at each step.

(b) Classification rates at each step.

(c) Detailed comparison of the number of stored support vectors at each step.

(d) Detailed comparison of the classification rates at each step.

Fig. 7. Average results of the knowledge transfer experiments when the updation was performed frames by frames. Fig. 7(a) & (b) compare the total number of stored support vectors and the final classification rates for all the four experiments. Fig. 7(c) & (d) present detailed results for one representative experiment.

become a possible source of misleading information. We tested our algorithm on two different scenarios, continuous learning of visual place models under dynamic changes and knowledge transfer across robot platforms. In both scenarios, we benchmarked our method against a recent approach, showing in both cases that our technique achieve the same accuracy while at the same time obtaining a significant reduction in the memory requirements.

Acknowledgments. This work was supported by EU project DIRAC (FP6-0027787).

REFERENCES

[1] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[2] N. Cristianini and J. S. Taylor., *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[3] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE TRA*, vol. 17, no. 3, pp. 229–241, 2001.

[4] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *Proc. of ICRA'07*.

[5] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

[6] Y. J. Lee and O. L. Mangasarian, "RSVM: Reduced support vector machines," in *Proc. of SDM'01*.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[8] J. Luo, A. Pronobis, and B. Caputo, "Svm-based transfer of visual knowledge across robotic platforms," in *Proc. of ICVS'07*.

[9] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt., "Incremental learning for place recognition in dynamic environments," in *Proc. of IROS'07*.

[10] I. Nourbakhsh, R. Powers, and S. Birchfield, "Dervish: An office navigation robot," *AI Magazine*, vol. 16, no. 2, pp. 53–60, 1995.

[11] F. Orabona, C. Castellini, B. Caputo, J. Luo, and G. Sandini, "Indoor place recognition using online independent support vector machines," in *Proc. of BMVC'07*.

[12] A. Pronobis, O. Martínez Mozos, and B. Caputo, "SVM-based discriminative accumulation scheme for place recognition," in *Proc. of ICRA'08*.

[13] I. Steinwart, "Sparseness of support vector machines," *Journal of Machine Learning Research*, vol. 4, pp. 1071–1105, 2003.

[14] N. A. Syed, H. Liu, and K. K. Sung, "Incremental learning with support vector machines," in *Proc. of IJCAI'99*.

[15] A. Tapus and R. Siegwart, "Incremental robot mapping with fingerprints of places," in *Proc. of IROS'05*.

[16] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, P. Jensfelt, and H. I. Christensen, "Towards robust place recognition for robot localization," in *Proc. of ICRA'08*.

[17] C. Wallraven, B. Caputo, and A. Graf., "Recognition with local features: the kernel recipe," in *Proc. of ICCV'03*.