IDIAP RESEARCH REPORT

Incremental Learning for Place Recognition in Dynamic Environments

Jie Luo [a] Andrzej Pronobis [a]
Barbara Caputo [b c] Patric Jensfelt [a]

IDIAP–RR 06-52

October 2006

submitted for publication

[a] NADA/CAS, KTH, Sweden -pronobis@nada.kth.se
[b] IDIAP - bcaputo@idiap.ch
[c] Ecole Polytechnique Fédérale de Lausanne (EPFL) - 1015 Lausanne (Switzerland)

IDIAP Research Institute www.idiap.ch
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 06-52

Incremental Learning for Place Recognition in Dynamic Environments

Jie Luo Andrzej Pronobis Barbara Caputo Patric Jensfelt

October 2006

submitted for publication

Abstract. Vision-based place recognition is a desirable feature for an autonomous mobile system. In order to work in realistic scenarios, a visual recognition algorithm should have two key properties: robustness and adaptability. This paper focuses on the latter, and presents a discriminative incremental learning approach to place recognition. We use a recently introduced version of the fixed-partition incremental SVM, which allows to control the memory requirements as the system updates its internal representation. At the same time, it preserves the recognition performance of the batch algorithm and runs online. In order to assess the method, we acquired a database capturing the intrinsic variability of places over time. Extensive experiments show the power and the potential of the approach.

1 Introduction

A fundamental requirement for an autonomous mobile system is the ability to localize itself within a known environment. Vision based systems have gained popularity recently, and several methods have been proposed using vision alone [1, 2, 3], or combined with more traditional range sensors, like laser and sonars [4, 5]. We see two main reasons behind this trend: (1) vision potentially offers more portable and cost effective solutions, as new mass markets for camera technology results in significantly reduced prices and increased performance; (2) vision can provide information unavailable to other sensors: for instance, it can provide semantic information on a scene through the understanding of its visual appearance. This would open various opportunities in terms of flexibility and use of contextual information.

Current research on vision-based localization systems faces several issues, of which robustness and adaptability are probably the most challenging. The system should be robust to many types of variations such as changes in illumination conditions, people moving around, or objects being used and moved. Moreover, the visual appearance of indoor environments changes continuously in time. This poses serious problems for recognition algorithms trained off-line on data acquired once and for all during a fixed time span. At the same time, when used on a robot, the system must run in real-time on hardware with limited processing and memory resources.

In our previous work [6], we presented a purely appearance-based model able to cope with illumination and pose changes, and we showed experimentally that it could achieve satisfactory performances when considering short time intervals between the training and testing data acquisition. Nevertheless, a room's appearance is doomed to change dramatically in time because it is used: chairs are pushed around, objects are taken in/out of drawers, furniture and paintings are added, or changed, or rearranged; and so forth. As it is not possible to predict a priori how a room is going to change, it is not possible to acquire beforehand training data representative of all its possible visual variations. Thus, the only possible strategy for a visual place recognition algorithm aiming to work in realistic settings is to update its internal representation in time, learning incrementally from the new data recorded during use.

In this paper, we focus on the ability of the recognition algorithm to adapt to the changes over a long period of time. We argue that adaptation should be performed incrementally, and the internal representation should be updated (rather then rebuilt from scratch) without the need to keep all the previously acquired training data in memory. Moreover, the updating process should gradually forget unnecessary information and keep the model compact, fast, and free from redundancy. To achieve these goals we applied several Support Vector Machine incremental learning algorithms [7, 8, 9] to the domain of visual place recognition. In order to test their effectiveness on this scenario, we extended the database used in our previous experiments [6] with new data acquired similarly, but after 6 months, using the same two mobile robot platforms. Extensive experiments clearly show the power of our approach, while illustrating the need for incremental solutions in real–time mobile robotics.

The rest of the paper is organized as follows: after a review of related literature (Section 2), Section 3 presents our approach to incremental place recognition as well as our algorithms of choice. Section 4 describes the scenario and database used during the experimental evaluation, whose results are reported in Section 5. The paper concludes with a summary discussion and possible directions for future work.

2 Related Works in Place Recognition

Incremental learning approaches had been so far mostly used for constructing the geometrical map, or the environment representation, online. E. Brunskill et al. [10] proposed a SLAM model using incremental PCA for simultaneous localization and mapping. A similar approach was used in the only work we are aware of that uses an incremental method in the context of place recognition. In [11] they used incremental PCA to update low-dimensional representations of images taken by a mobile robot as it moved around in an environment. They also tested repetitive learning of their model with the same training images several times. Note that their work was not addressing the problem of environment variations and of its complexity in real-world data.

3 Incremental Place Recognition

This section describes our approach to incremental place recognition and the corresponding algorithm. We adopt the appearance-based paradigm, and we assume that a realistic scene can be represented by a global descriptor without any loss of discriminative information. Thus, as each learning step, every room is represented by a set of feature vectors, extracted by a collection of images capturing its visual appearance. The system updates incrementally its decision function: after having acquired a fixed, pre-defined number of images for each room, the algorithm triggers the incremental learning function and integrates potential new information in the existing internal representation. This leads to a visual place recognition system able to adapt in time to the natural changes of a real-world setting.

The rest of this section describes the basic principles of support vector machines (Section 3.1), two popular incremental version of the basic algorithm (Section 3.2) and our modified, memory-controlled version of incremental SVMs (Section 3.3). A comprehensive description of the experimental setup is given in Section 4.

3.1 Support Vector Machines

Support Vector Machines (SVMs, [12, 13]) belong to the class of large margin classifiers. Consider the problem of separating the set of training data $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots (\boldsymbol{x}_m, y_m)$ into two classes, where $\boldsymbol{x}_i \in \Re^N$ is a feature vector and $y_i \in \{-1, +1\}$ its class label (for the multi-class extensions, we refer the reader to [12, 13]). If we assume that the two classes can be linearly separated, the optimal hyperplane is the one which has maximum distance to the closest points in the training set, resulting in a classification function

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i \cdot \boldsymbol{x} + b\right), \tag{1}$$

where α_i and b are found by using an SVC learning algorithm [12, 13]. Most of the α_i's take the value of zero; $\boldsymbol{x}_i$ with nonzero α_i are the "support vectors". A nonlinear SVM can be constructed by replacing the inner product $\boldsymbol{x} \cdot \boldsymbol{y}$ in the linear SVM by the kernel function $K(\boldsymbol{x}, \boldsymbol{y})$ [12, 13]

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right). \tag{2}$$

This corresponds to constructing an optimal separating hyperplane in the feature space. In this paper we use the χ^2 kernel [14] $K(\boldsymbol{x}, \boldsymbol{y}) = \exp\{-\gamma\chi^2(\boldsymbol{x}, \boldsymbol{y})\}$, which has shown to give good performances for histogram-like features [6] for place recognition.

3.2 SVM: an Incremental Extension

Among all incremental SVM extensions proposed in the machine learning literature so far [7, 8, 15], approximate methods seem to be the most suitable for visual recognition: firstly - as opposed to exact methods like [15]- they discard a significant amount of the training data at each incremental step. Secondly, they are expected to achieve performances not too far from those obtained by an SVM trained on the complete data set (batch algorithm), because at each incremental step the algorithm remembers the essential class boundary information regarding the data seen so far (in form of support vectors). This information contributes properly to generate the classifier at the next iteration.

Once a new batch of data is loaded into memory, there are different possibilities for the updating of the current model, which might discard a part of the new data according to some fixed criteria

[8, 7]. In this paper we used two methods, the fixed-partition [7] and the error-driven technique [8].
Fixed-partition technique In this method the training data set is partitioned in batches of fixed size k:

$$\boldsymbol{T} = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_m, y_m)\} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, \dots \boldsymbol{T}_n\},$$

with $\boldsymbol{T}_i = \{(\boldsymbol{x}^i_j, y^i_j)\}^k_{j=1}$. At the first step, the model is trained on the first batch of data $\boldsymbol{T}_1$, obtaining a classification function

$$f_1(\boldsymbol{x}) = \mathrm{sgn}\left(\sum_{i=1}^{m_1} \alpha^1_i y^1_i \boldsymbol{x}^1_i \cdot \boldsymbol{x} + b^1\right).$$

At the second step, a new batch of data is loaded into memory; then, the *new* training set becomes

$$\boldsymbol{T}^{inc}_2 = \{\boldsymbol{T}_2 \cup \boldsymbol{S}V_1\}, \qquad \boldsymbol{S}V_1 = \{\boldsymbol{x}^1_i\}^{m_1}_{i=1},$$

where $\boldsymbol{S}V_1$ are the support vectors learned at the first step. The new classification function will be:

$$f_2(\boldsymbol{x}) = \mathrm{sgn}\left(\sum_{i=1}^{m_2} \alpha^2_i y^2_i \boldsymbol{x}^2_i \cdot \boldsymbol{x} + b^2\right).$$

Thus, as new batches of data points are loaded into memory, the existing support vector model is updated, so to generate the classifier at that incremental step.
Error-driven technique As opposed to the method described above, the error-driven technique makes a filtering on the batch data at each incremental step: given the model SVM_t at the step t, the new batch data are loaded into memory and classified using SVM_t. If the data is misclassified it is kept, otherwise it is discarded. Once a pre-fixed number n_e of misclassified data is collected, the support vector of SVM_t, together with the n_e misclassified points, are used as training data to obtain the new model SVM_{t+1}.

A common problem to both these approaches, and in general to all incremental extensions of SVM, is that in principle there is no limitation to the memory growth. Indeed, several experimental evaluations shows that, while approximate methods generally achieve classification performances equivalent to those of batch SVM, the number of SVs tends to grow proportionally to the number of incremental steps [9]. This is of course a serious issue for a method designed to work on a robotic platform.

3.3 Memory-Controlled Incremental SVM

In [9] we proposed a modification of the fixed-partition algorithm which leads to a controlled growth of the memory requirements for incremental SVM. Experiments on collections of images showed promising results. Here we propose to use this method on image sequences, so to obtain an adaptive model. The core idea of our memory-controlled algorithm is that the set of support vectors $\boldsymbol{X} = \{\boldsymbol{x}_i\}^m_{i=1}$ in eq (2) is not guaranteed to be linearly independent. Based on this observation, it is possible to reduce the number of support vector of a trained classifier, eliminating those which can be expressed as a linear combination of the others in the feature space. By updating the weights accordingly, it is ensured that the decision function is exactly the same as the original one [16]. More specifically, let us suppose that the first r support vectors are linearly independent, and the remaining $m-r$ depend linearly on those in the feature space: $\forall j = r+1, \dots m$, $\boldsymbol{x}_j \in span\{\boldsymbol{x}_i\}^r_{i=1}$. Then it holds

$$K(\boldsymbol{x}, \boldsymbol{x}_j) = \sum_{i=1}^{r} c_{ij} K(\boldsymbol{x}, \boldsymbol{x}_i). \tag{3}$$

By substituting eq (3) into eq (2) and re-defining the weights (we refer the readers to [16, 9] for the detailed derivation), we get:

$$f(\boldsymbol{x}) = \mathrm{sgn}\left(\sum_{i=1}^{r} \widehat{\alpha}_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b\right) \tag{4}$$

Figure 1: Sample images illustrating the variations captured in the IDOL2 database. Images in the top row show the variability introduced by changes in illumination for two rooms (first six images) as well as people appearing in the environment. The middle row shows the influence of people's everyday activity (first four images) as well as larger variations which happened over a time span of 6 months. Finally, the bottom row illustrates the changes in viewpoint observed for a series of images acquired one after another in 1.6 second.

with

$$\widehat{\alpha_i} = \alpha_i(1+\gamma_i) = \alpha_i\left(1 + \sum_{j=r+1}^{m} \frac{\alpha_j y_j c_{ij}}{\alpha_i y_i}\right). \tag{5}$$

Thus, the resulting classification function (eq. (4)) requires now $m-r$ less kernel evaluations than the original one. Note that the number r of linearly independent vectors depends on the definition of linear independence given by the reduction algorithm, which ultimately is controlled by a threshold value τ [9]. Thus, this parameter can be effectively used to trade performance for memory requirements and speed during classification, depending on the task at hand.

By combining this algorithm with the approximate techniques, it is possible to obtain an incremental SVM with a mechanism which reduces the memory growth in a principled way. We apply the reduction scheme at each incremental step; thus, the new representation of the data is built from the remaining support vectors. Experiments, reported in Section 5, show that this technique is an effective solution for a controlled growth of the memory requirements.

4 Experimental Setup

This section describes the setup used for the experiments reported in this paper. We first briefly describe the experimental scenario (section 4.1), and then we introduce a new database for visual place recognition from robotic platforms named IDOL2 (section 4.2).

4.1 Experimental Scenario

The presented incremental SVM algorithms allow to update the existing internal representation without the need to rebuild it from scratch. Beside this fundamental ability, we have identified several other desirable properties for incremental learning system running on robotic platforms: (a) *Reduced Memory Requirements.* Once the algorithm receives a new batch of data, it should update its internal model and then discard all the unnecessary data. The model representation of incremental SVM algorithms is given by the weighted linear combination of a discriminant subset of the training vector; thus, it is important to verify, at each incremental step, the model capability to discard the highest possible amount of vectors without any significant loss of performance with respect to the batch algorithm. Experiments investigating this property are reported in section 5.1 and 5.2. (b)

Forgetting Mechanism. As places change their visual appearance in time, the algorithm should accordingly change its internal representation and privilege newest information at the expenses to oldest one. For incremental SVM, this means that the algorithm should tend to discard support vectors coming from previous incremental steps, and build its model at each iteration on a significant fraction of the newest input data. We are not aware of experimental studies on this aspect of incremental extensions of SVMs; experiments addressing this issue are described in section 5.1. *(c) Self-Detection of Knowledge/Ignorance.* Regardless of the algorithm complexity, updating the internal representation at every incremental step is computationally expensive. Ideally, the algorithm should be able to detect if a new set of data contains enough new information to justify the computational expense of updating the model. We are not aware of any incremental model able to perform these steps, and incremental SVM is no exception. Still, as a first step towards this goal, we have identified the reaching of a plateau in memory growth and in recognition rate as quantitative indicators which might be used in the future for implementing these functions. Experimental evaluation of these behaviors is reported in section 5.1 and 5.2. *(d) Speed.* Last but not least, the incremental update should be done on-line, thus the algorithm should have the lowest possible algorithm complexity. An analysis on this aspect for incremental SVMs is given in section 5.3.

The design of experiments investigating all these points requires an ad-hoc database. We thus extended the image sequences we acquired for [6], adding image sequences as well as higher variability into the rooms over long spans. Its detailed description is the topic of the next section.

4.2 The IDOL2 Database

The IDOL2 (Image Database for rObot Localization 2, [17]) database contains 24 image sequences acquired by a perspective camera, mounted on two mobile robot platforms Minnie and Dumbo. The robots were manually driven through an indoor laboratory environment and the images were acquired at a rate of 5fps. On Minnie the camera's height was 98cm above the floor, whereas on Dumbo it was 36cm. Each image sequence consists of 800-1100 frames automatically labeled with one of five different classes (printer area, corridor, kitchen, two persons office and one person office). The labeling is based on the camera position given by laser-based localization system. The acquisition procedure was designed to capture the changes in illumination and varying weather conditions (sunny, cloudy and night). Also, special care was taken to capture people's activities, change of location for objects and for furniture; for one environment (the two-persons office room) we were able to record a quite dramatic change in decoration, which happened over a long time span (6 months). Figure 1 shows some sample images from the database, illustrating these variations.

The 24 image sequences are divided as follows: for each robot platform and for each weather condition, we recorded 4 sequences. Of these four sequences, the first two were acquired six months before the last two. This means that, for every robot and for every illumination condition, we always have two sequences acquired under similar conditions, and two sequences acquired under very different conditions. This makes the database useful for several types of experiments. It is important to note that, even for the two sequences acquired within a short time span, variations still exist from everyday activities and viewpoint differences during acquisition. For further details, we refer the reader to [17].

In order to test the various properties of interest of the incremental algorithms, we needed a reasonable number of incremental steps. Thus, we splitted every sequence into 5 subsequences, so that each subset contained one of the five images acquired by the robot every second. Since during acquisition the camera's viewpoint changes, the subsequences could be considered as recorded separately in a static environment but for varying pose. In order to get a feeling of the variations of the frame images in a sequence, bottom row of Figure 1 shows some sample images acquired within a time span of 1.6 sec.

5 Experimental Results

We conducted two series of experiments to evaluate the effectiveness of our approach. In all the experiments, we compared the three incremental techniques as well as the batch algorithm.

The evaluation was performed using composed receptive field histograms (CRFH) [18] as global image features[1]. We built the histograms from first order normalized Gaussian derivative filters applied to the images at two different scales, and we used χ^2 as a kernel for SVM. Such combination previously proved effective for the place recognition task [6]. For all the experiments, we employed our extended version of the *libsvm* [21] library, and we determined the SVM and kernel parameters via cross validation. For the memory-controlled incremental algorithm, the threshold parameter was adjusted so to allow, at most, a reduction in recognition rate of 1% of that obtained with the fixed-partition method.

5.1 Examining properties of the incremental methods

In the first series of experiments, the system was trained incrementally on three sequences acquired under similar illumination conditions with the same robot platform; the fourth sequence was used for testing. Training on each sequence was performed in 5 steps, using one subsequence at a time, resulting in 15 steps in total. We considered 36 different permutations of training and test sequences; here we report average results with standard deviations. Fig. 2, top, shows the recognition rates obtained at each step using the three incremental algorithms (fixed-partition, error-driven, and memory-controlled) as well as the batch method on the whole training data. Fig 2, bottom, reports the number of support vectors stored in the model at each step of the incremental procedure. We can observe that the fixed-partition incremental algorithm requires less support vectors than the batch one, while achieving an identical performance. Also, both algorithms show plateaus in the classification rate whenever the model is trained on similar data, coming from consecutive subsequences. This behaviour is not reflected in the size of the model: for both techniques, the number of support vectors grows continuously with the number of training step. This behavior would eventually lead to a memory explosion, and it makes us conclude that the batch and the fixed-partition incremental algorithms are not suitable for this application.

The other two incremental extensions (memory-controlled and error-driven) seem to be better suited for continuous learning. We see that for these methods, both the classification rate and the number of stored support vectors show plateaus every five incremental steps (Fig 2, top and bottom). The error-driven technique is the model with the smallest memory growth and requirement; however, it also delivers the worst recognition performance. At the same time, the memory-controlled algorithm performs comparably to the batch SVM, with memory requirements twice smaller. Furthermore, the memory growth slows down over time (Fig 2, bottom). We therefore conclude that the memory--controlled incremental SVM is the best algorithm, of the four considered, for the problem at hand.

In order to gain a better understanding of the methods' behavior, we performed an additional analysis of the results. Fig 3e shows, for the three incremental techniques, the average amounts of vectors (originating from each of the three training sequences) that remained in the model after the final incremental step. The figure illustrates how the methods weight instances, learned at different time, when constructing the internal representation. We see that both fixed-partition and memory-controlled algorithms privileges new data, as the support vectors from the last training sequence are more represented in the model. This phenomenon is stronger for the memory-controlled algorithm, while it is not shown by the error-driven method, which seems more conservative.

To get a feeling of how the forgetting mechanism works in case of the memory-controlled method, we plotted the positions where the support vectors were acquired. Fig. 3 reports results obtained for

[1] The experiments were conducted also for local image features. We used SIFT [19] as local descriptor and local kernels [20] for SVM. The experimental findings are similar to those reported here, and thus are omitted for space reasons.

a model built after the final incremental step. The positions were marked on three maps presented in Fig. 3a,b,c so that each of the maps shows the support vectors originating from only one training sequence. As already shown in Fig. 3e, most of the vectors in the model come from the last training sequence. Moreover, the number of SVs from the previous training steps decreases monotonically, thus the algorithm gradually forgets the old knowledge. It is interesting to observe how the vectors from each sequence distribute along the path of the robot. On each map, the places crowded with SVs are mainly transition areas between the rooms, regions of high variability, as well as places at which the robot rotated (thus providing a lot of different visual cues without changing position). To illustrate the point, Fig. 3d shows sample images acquired in the corridor, for which the support vectors decay quickly, and one of the offices, for which they are being preserved much longer. The results indicate that the forgetting is not performed in a random way. On the contrary, the algorithm tends to preserve those training vectors that are most crucial for discriminative classification.

5.2 Real-world experiment

The next step was to test all the incremental methods in a real-world scenario. For this purpose, we considered a case where the algorithms had to incrementally gain robustness to variations, introduced by changing illumination and natural activity, and they also had to use their adaptation abilities to handle long-time environment changes. We first trained the systems on three sequences acquired at roughly similar time, but under different illumination conditions. Then, we repeated the same training procedure on sequences acquired 6 months later. In order to increase the number of incremental steps, and differentiate the amount of new information introduced by each set of data, each sequence was again divided into five subsequences. In total, for each experiment we performed 30 incremental steps. Since the IDOL2 database consists of pairs of sequences acquired under roughly similar conditions, each training sequence had a corresponding sequence which could be used for testing.

The experiment was repeated 12 times for different orderings of the training data. Fig 4 reports the average results together with standard deviations. Fig. 4a, compares the amounts of SVs stored in the models at each incremental step for all the methods. The first column in Fig 4 b,c and d shows the recognition performances, obtained by the three methods, on 6 different test sequences. We can observe that (a) once again, the fixed-partition and the memory controlled algorithms perform similarly, and better than the error-driven one; (b) all the approaches show a similar behavior: they perform well on the first testing sequence (that is the one most similar to the first training sequence, from which the models are built) while performance is poorer for the other 5 testing sequences. Thus, after having seen only one sequence, the systems seem not to be able to generalize. If one observes the second column in Fig Fig 4 b,c and d, the two trends described above are confirmed (i.e. the methods perform well on two test sequences, after having being trained on two similar one). But if one compares the results between the first and the second column, for all the methods, it shows that all algorithm have an increase in performance on the other testing sequences. thus, the models gained robustness by adapting incrementally the internal representations, and incorporating partly the new available information. This behavior is present for all the columns, and all the methods. These results illustrate how the learning process affects the performance, and emphasize the need for adaptation. As a final comment, we remark that also for these experiments the memory-controlled algorithm provided the best trade-off between performance and memory requirements.

5.3 Discussion

The presented results provide a clear evidence of the capability of the discriminative methods to perform incremental learning for vision-based place recognition, and their adaptability to variations in the environment. For all experiments, we found that the fixed-partition method performs as batch SVM, but it is unable to control the memory growth. We also found that the error-driven method could get a reasonable accuracy while minimizing the memory requirements. Still, even if for this last algorithm the number of stored SVs increases slowly, the growth is anyway not predictable neither controllable. Moreover, none of these two methods has shown to possess a forgetting mechanism. On the contrary, experiments have shown that they tend to prefer, when building their decision functions, data acquired during the first training steps. In this sense they can be described as conservative, and we can expect that they would adapt slowly in highly dynamic environments. As opposed to this, the memory-controlled algorithm is able to achieve performances statistically equivalent to those of batch SVM, while at the same time providing a principled and effective way to control the memory growth. Experiments showed that this has induced a forgetting mechanism which privileges newly acquired data to the expenses of oldest one, while reaching a memory plateau whenever new data are similar to those already processed. All the incremental methods are intrinsically on-line. They allow the system to delete the training images immediately after updating the model. Since it separates the training task, the training time becomes eventually lower, than the batch learning. In the second experiment, we observe that the last step's training time dropped from the batch method's 25.48s to memory-reduced method's 5.61s on a 2.6GHZ Pentium IV machine when trained on 5635 images from 6 sequences. The prediction per frame is more than 200% faster because of the condensed size of the recognition model.

6 Summary and Conclusion

We proposed a discriminative incremental learning approach to place recognition, using a version of incremental SVM, which allows to control the memory growth as the system keeps acquiring new data. Extensive experiments show that our method achieves recognition performances statistically equivalent to those of the batch algorithm, while obtaining a dramatic memory reduction. Moreover, we showed experimentally that (a) the method tends to forget the oldest support vectors in favor of newest data when updating the decision function, and (b) it reaches a plateau in performance and memory whenever it is presented with data sequences very similar to those already learned. This seems to indicate that our algorithm can "recognize" if a new set of data contains novel information or not. We plan in the future to translate this plateau behavior into measurable quantities, and to use them for switching on/off the incremental update. This would lead to an algorithm that modifies its internal representation only in presence of new information. Other issues we plan to investigate are the scalability of the approach and the possibility to do adaptation room by room, as the new data is acquired.

References

[1] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *Proc. ICRA'01.*

[2] H. Tamimi and A. Zell, "Vision based localization of mobile robots using kernel approaches," in *Proc. IROS'04.*

[3] I. Ulrich and I. R. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proc. ICRA'00.*

[4] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," in *Proc. of the National Conference on Artificial Intelligence (AAAI-94)*, 1994.

[5] A. Tapus and R. Siegwart, "Incremental robot mapping with fingerprints of places," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, Aug. 2005, pp. 172–177.

[6] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, "A discriminative approach to robust visual place recognition," in *Proc. IROS'06.*

[7] H. L. N. A. Syed and K. K. Sung, “Incremental learning with support vector machines,” in *Proc. IJCAI'99*.

[8] C. Domeniconi and D. Gunopulos, “Incremental support vector machine construction,” in *Proc. ICDM'01*.

[9] A. Pronobis and B. Caputo, “The more you learn, the less you store: memory-controlled incremental support vector machines, Tech. Rep. IDIAP-RR-06-51, 2006, available at `http://www.idiap.ch/publications.php`.

[10] E. Brunskill and N. Roy, “Slam using incremental probabilistic pca and dimensionality reduction,” in *Proc. ICRA'05*.

[11] M. Artač, M. Jogan, and A. Leonardis, “Mobile robot localization using an incremental eigenspace model,” in *Proc. ICRA'02*.

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

[13] V. Vapnik, *Statistical learning theory*. New York: Wiley and Son, 1998.

[14] S. Belongie, C. Fowlkes, F. Chung, and J. Malik, “Spectral partitioning with indefinite kernels using the nyström extension,” in *Proc. ECCV'02*.

[15] G. Cauwenberghs and T. Poggio, “Incremental and decremental support vector machine learning,” in *Proc. NIPS'00*.

[16] T. Downs, K. E. Gates, and A. Masters, “Exact simplification of support vector solutions,” *J. Mach. Learn. Res.*, vol. 2, 2002.

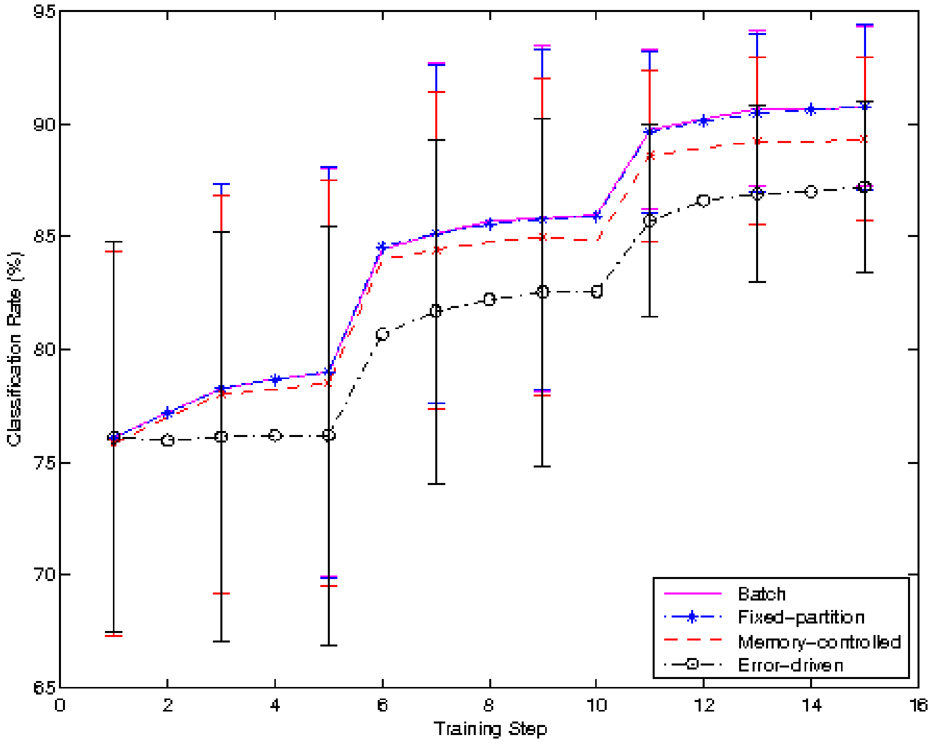
[17] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt, “The IDOL2 database,” KTH, CAS/CVAP, Tech. Rep., 2006, available at `http://www.nada.kth.se/cvap/databases/kth-idol2`.

[18] O. Linde and T. Lindeberg, “Object recognition using composed receptive field histograms of higher dimensionality,” in *Proc. ICPR'04*.

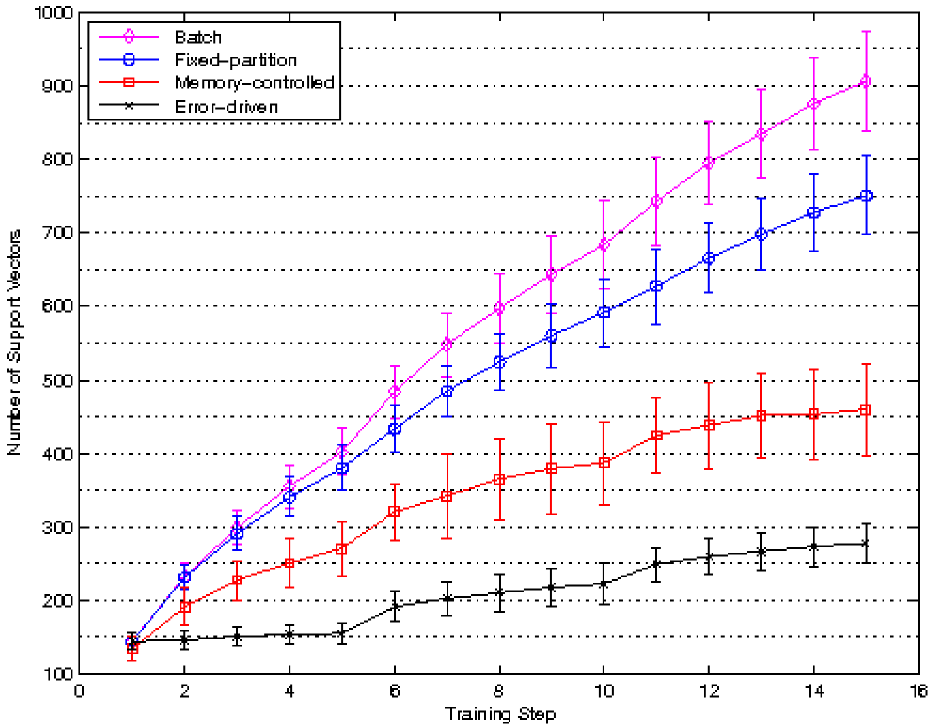
[19] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proc. of ICCV'99*.

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

[21] C. C. Chang and C. J. Lin, *LIBSVM: A Library for Support Vector Machines*, 2001, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.



(a) Classification rate at each incremental step.



(b) Number of support vectors at each incremental step.

Figure 2: Average results obtained for experiment performed on sequences acquired under similar illumination conditions with the same robot platform for three incremental methods and batch algorithm.

(a) 78 SVs from the first training sequence.

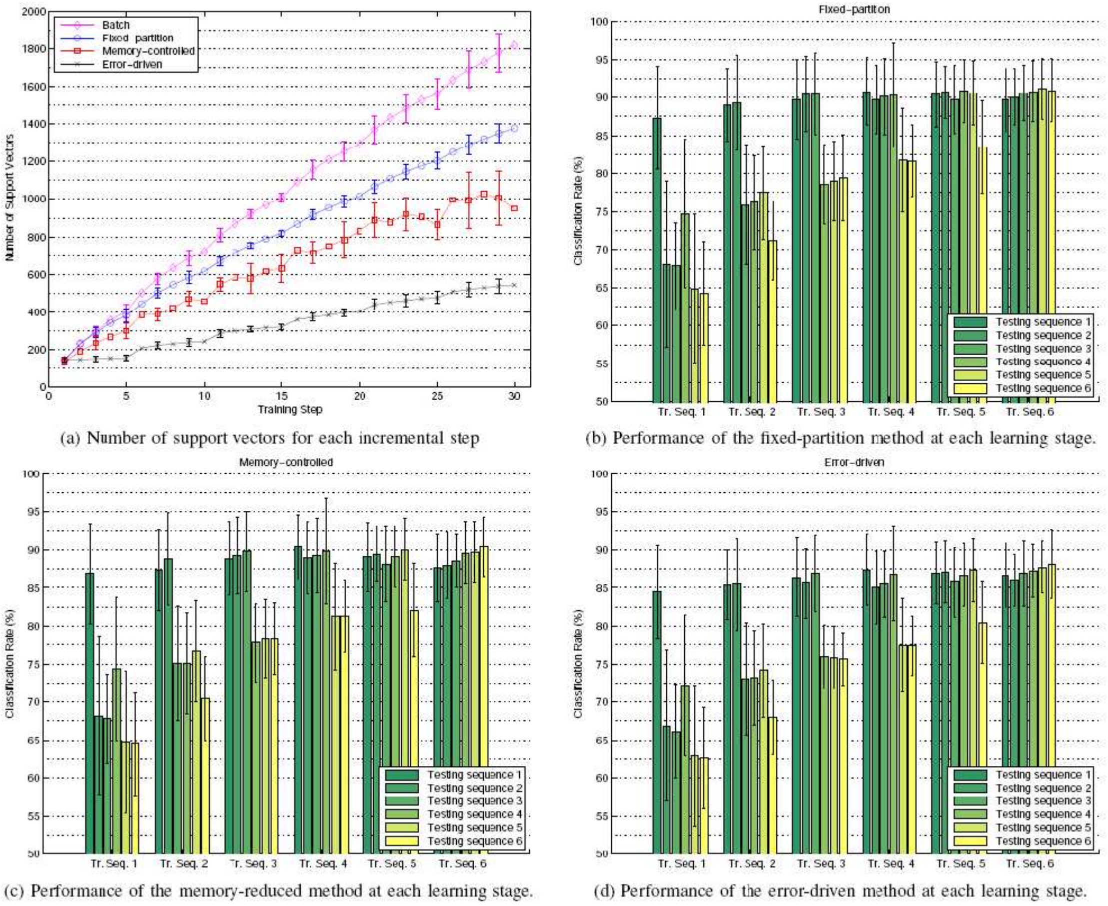
(b) 111 SVs from the second training sequence.

(c) 149 SVs from the third training sequence.

(d) Sample images from the three training sequences.

(e) Statistics of SVs stored in the final incremental models.

Figure 3: Maps of the environment with plotted positions of the support vectors stored in the model obtained after the final incremental step for one of the experiments conducted using the memory-controlled technique. The support vectors were divided into three maps (a, b, and c) according to the training sequence they originate from. Additionally, each map shows the path of the robot during acquisition of the sequence (arrows indicate the direction of driving). We observe that the SVs from the old training sequences were gradually eliminated by the algorithm and this effect was stronger in regions with lower variability. Sample images captured in regions of different variability can be seen in Fig. 3d. Fig. 3e compares the average amounts of training vectors coming from the three sequences that were stored in the final incremental model for all the three considered incremental techniques.



(a) Number of support vectors for each incremental step

(b) Performance of the fixed-partition method at each learning stage.

(c) Performance of the memory-reduced method at each learning stage.

(d) Performance of the error-driven method at each learning stage.

Figure 4: Average results of the incremental experiments in a real-world scenario. Fig. 4a shows the number of support vectors stored in the model after each incremental step for the three incremental techniques and the batch method. Fig. 4b,c,d present the classification rates obtained by testing the models built after every fifth incremental step (after the classifier have been trained on a full sequence) with all the available test sets.