SVM-based Transfer of Visual Knowledge Across Robotic Platforms

Jie Luo[1,2], Andrzej Pronobis[3], and Barbara Caputo[1,2]

[1] IDIAP Research Institute, Rue du Simplon 4, Martigny, Switzerland
[2] EPFL, 1015 Lausanne, Switzerland
[3] Centre for Autonomous Systems, Royal Institute of Technology, Stockholm, Sweden
{jluo,bcaputo}@idiap.ch pronobis@nada.kth.se

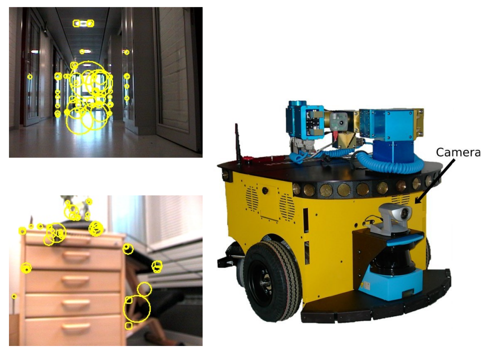
Abstract. This paper presents an SVM-based algorithm for the transfer of knowledge across robot platforms aiming to perform the same task. Our method exploits efficiently the transferred knowledge while updating incrementally the internal representation as new information is available. The algorithm is adaptive and tends to privilege new data when building the SV solution. This prevents the old knowledge to nest into the model and eventually become a possible source of misleading information. We tested our approach in the domain of vision-based place recognition. Extensive experiments show that using transferred knowledge clearly pays off in terms of performance and stability of the solution.

1 Introduction

The ability to transfer knowledge between different domains enables humans to learn efficiently from small number of examples. This observation inspired robotics and machine learning researchers to search for algorithms able to exploit prior knowledge so to improve performance of artificial learners and speed up the learning process. In this paper we investigate the problem of transfer of visual knowledge between robotic platforms with different characteristics, engaged in the same task i.e. vision-based place recognition. We consider a scenario where a robot, proficient in solving the task within a known environment, transfers its knowledge to another robotic platform, which is a tabula rasa. To tackle this problem, it is necessary an efficient way of exploiting the knowledge transferred from a different platform as well as updating the internal representation when new training data are available. The knowledge transfer scheme should be adaptive and privilege newest data so to prevent from accumulating outdated information. Finally, the solution obtained starting from a transferred model should gradually converge to the one learned from scratch, not only in terms of performance on a task but also of required resources (e.g. memory). This is particularly important when the algorithm is to be used on a robot.

The problem of knowledge transfer is well known in the robotic and machine learning communities. Thrun and Mitchell [1, 2] 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 the problem have also been made from the perspective of Reinforcement Learning, including the case of transferring learned skills between different RL agents [3, 4].

(a) Robot A



(b) Robot B

Fig. 1. The two mobile robot platforms used for image acquisition as well as pairs of images acquired with both devices at a similar pose. The interest points detected in the images using the Harris-Laplace detector are marked with yellow circles, and the radius of each circle indicates the scale at which the point was detected.

This paper focuses on an algorithm that allows to perform transfer of knowledge represented in form of a model trained using the Support Vector Machines. The algorithm was thoroughly tested in the domain of vision-based place recognition, where the knowledge was exchanged between two mobile robot platforms. Since it is desirable to perform an update of the internal representation as soon as new data are available, we investigated the behavior of the algorithm when the model was trained using a small number of examples, and only one class at a time. Our experimental evaluation showed that the system exploits successfully the prior knowledge, obtaining a remarkable boost in performance compared to a model built from scratch in analogous conditions.

This paper is organized as follows: Section 2 gives a definition of the problem and discusses the main issues related to the transfer of knowledge. Then, Section 3 presents our approach. Sections 4 and 5 describe the experimental setup and report results of our experimental evaluation. Conclusions are drawn in Section 6.

2 Problem Statement

Our focus is on a transfer of visual knowledge between two agents having different characteristics. Specifically, we consider the case where knowledge related to the problem of visual-based place recognition is transferred from Robot A to Robot B. The main difference between the two platforms lies in the height of the cameras (see Fig. 1). Both robots are engaged in the same task and operate in the same environment. Their recognition system is based on the SVM classifier, thus they share the same knowledge representation. The aim is to efficiently exploit the knowledge acquired e.g. by Robot A so to boost the recognition performance of Robot B. Obtaining robustness to visual variations in a complex scenario can be a costly process; thus, it is of great importance the ability of exchanging skills that, once learned, could be valuable to other devices operating in the same environment.

The challenges in the transfer of knowledge will come from:
(a) Differences in the parameters of the two platforms The cameras are mounted at two different heights, thus the informative content of the images acquired by the two platforms is different (see Fig. 1). Because of this, the knowledge acquired by one platform might not be helpful for the other one or, in the worst case, it might constitute an obstacle.
(b) Room by room/frames by frames knowledge update The model from Robot A should be adapted to the needs of Robot B as soon as new data are available. This can be done in a room-by-room (class-by-class), or frames-by-frames fashion; both scenarios are at risk of unbalanced data with respect to the class being updated. Note that these scenarios differ from the "off-line" incremental learning scenario described in [5], where the robot updates its model after having acquired data from the whole environment.
(c) Growing memory requirements In case of SVMs, the model is parametrized by a subset of training samples. As a result, building on top of an already trained classifier might lead to a solution that will be much more demanding in terms of memory usage and computational power than the one learned from scratch [5].

3 The Algorithm

The following section provides a short description of the place recognition system used as a framework for our experiments and describes our approach to the transfer of knowledge problem. The place recognition system implemented on both mobile robots is based on the Support Vector Machine classifier, combined with local image features computed using a Harris-Laplace detector [6] and the SIFT descriptor [7]. The local descriptors are used as input to SVM via the match kernel presented in [8]. Preliminary experiments showed that the local features are more suitable for the transfer of knowledge in our scenario than global features, like composed receptive field histograms [9], which were successfully used for robust place recognition in a similar framework [10]. This is primarily due to the fact that the local features are generally more robust to occlusions and viewpoint changes which is a desirable property in our setting.

The knowledge (i.e. the model) of the SVM classifier is given in form of a set of support vectors, their corresponding Lagrange multipliers α_i, and a bias factor b [11]. This model representation is the same for both robots, which make the transfer of knowledge possible. Once the knowledge from the platform Robot A is loaded into the memory of the platform Robot B, the system begins to update this support vector model using the data acquired by its own sensor. This update procedure is based on the fixed-partition incremental SVM algorithm [12]. More specifically, the decision function transferred from Robot A to Robot B will be

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

where $\boldsymbol{x}_i$ denotes support vectors and $y_i \in \{-1, +1\}$ their class labels (for multi-class extensions we refer the reader to [11]). As Robot B is shown the environment, it will

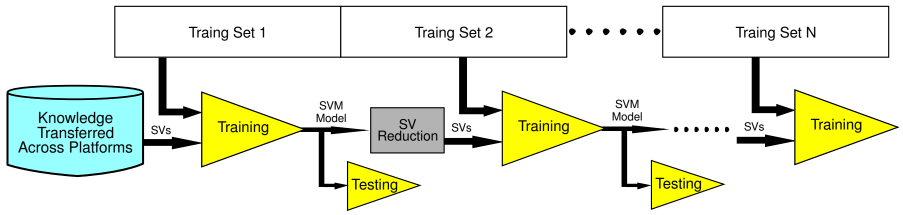


Fig. 2. A diagram illustrating the data flow in the knowledge-transfer system.

acquire consecutive batches of data

$$\boldsymbol{T}_k^B = \left\{(\boldsymbol{x}_j^B, y_j^B)\right\}_{j=1}^N .$$

Each batch of data corresponds to a set of N labeled feature vectors extracted from a part of an image sequence. According to the fixed-partition incremental SVM method [12], the model update is performed by retraining the classifier on support vectors from the previous model combined with new training data. Thus, the training data for the first model update will be

$$\boldsymbol{T}_1 = \{\boldsymbol{T}_1^B \cup \boldsymbol{SV}^A\}, \qquad \boldsymbol{SV}^A = \{(\boldsymbol{x}_i^A, y_i^A)\}_{i=1}^{M_A},$$

where $\boldsymbol{SV}^A$ are the support vectors of the decision function (1), i.e. the transferred knowledge from Robot A. The new classification function will be:

$$f^B(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{\widehat{M}_A} \widehat{\alpha}_i^A y_i^A \boldsymbol{x}_i^A \cdot \boldsymbol{x} + \sum_{i=1}^{M_B} \alpha_i^B y_i^B \boldsymbol{x}_i^B \cdot \boldsymbol{x} + b^B\right), \quad \widehat{M}_A \leq M_A. \tag{2}$$

The same procedure is followed for every successive update of the model used by Robot B.

As the system keeps updating the transferred representation, it adapts the model to its own perception of the environment. It would be desirable that the system could progressively substitute support vectors from the knowledge model of Robot A with support vectors acquired by Robot B. To promote this behavior, while at the same time reducing the memory requirements, we applied, after each incremental update, the support vector reduction algorithm proposed by Downs et al [13]. The method consists in identifying the support vectors which are linearly independent, and rewrite the SVM decision function as a linear combination of those vectors, and those vectors only. The Lagrange multipliers are recomputed accordingly, achieving a reduction in the number of support vectors while preserving the exact solution. Fig. 2 shows our framework for transfer of knowledge. In order to privilege information coming from the platform currently in use, we imposed to the algorithm to discard only the support vectors that were linearly dependent *and* came from the previous platform. This scheme speeds up the turnover of stored support vectors, while preferring newest data and at the same time preserving relevant information. Thus, our approach favors adaptation while reducing the memory requirements, and discards outdated knowledge which might otherwise act as noisy information.

Fig. 3. Pictures taken from the IDOL2 database illustrating the appearance of the five rooms from the point of view of both robotic platforms.

4 The Database

For our experiments we used the IDOL 2 database (Image Database for rObot Localization 2, [14]) which contains 24 image sequences acquired using a perspective camera mounted on two mobile robot platforms. The acquisition was performed within an indoor environment consisting of five rooms of different functionality: One-person Office (OO), Two-persons Office (TO), CoRridor (CR), KiTchen (KT) and Printer Area (PA). The sequences were acquired under various weather and illumination conditions (sunny, cloudy, and night) and across a time span of six months. Thus, this data capture natural variability that occurs in real-world environments because of both natural changes in the illumination and human activity. Note that the focus of our work here is to study how to transfer structured knowledge across platforms, rather than handling different types of visual variations. Our choice of the database is thus purely due to its dimension, which allowed us to test extensively our approach. For further details on the database, we refer the readers to [14].

Both mobile robot platforms, the PeopleBot Robot A and the PowerBot Robot B, are equipped with the pan-tilt-zoom Cannon VC-C4 camera. As shown in Fig. 1a&b, the cameras are mounted at different heights. On Robot A the camera is located 98*cm* above the floor, whereas on Robot B its height is 36*cm*. Furthermore, the camera on Robot B is tilted up 13°, so to reduce the amount of floor captured in the images. Fig. 1 shows some sample images from the database acquired by both robots from very close viewpoints, illustrating the difference in visual content. These images were acquired under the same illumination conditions and within short time spans.

The image sequences in the database are divided as follows: for each robot platform and for each type of illumination conditions, there were four sequences recorded. Of these four sequences, the first two were acquired six months before the last two. This means that, for each robot and for every illumination condition, we always have two sequences acquired under similar conditions, and two sequences acquired under very different conditions. In all our experiments, we will always use those sequences acquired under similar conditions, one as training set and the another one as test set.

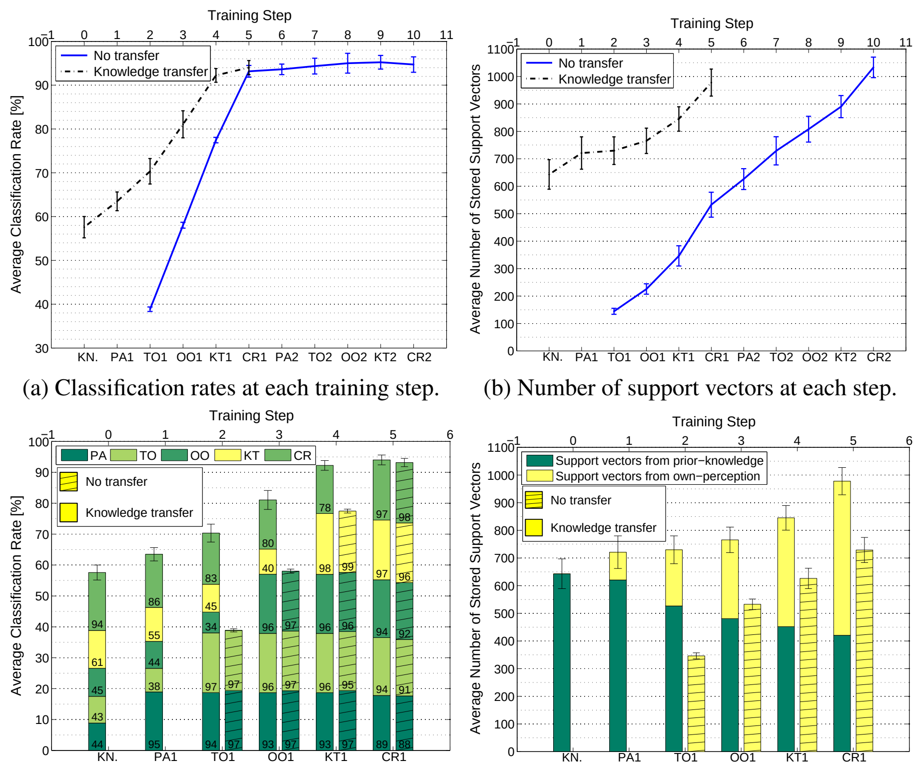
5 Experiments

We conducted two series of experiments to evaluate our system. In all the experiments, we benchmarked against a system not using any prior knowledge. The evaluation was performed using our extended version of the *libsvm* library [15]. The SVM and kernel parameters were determined via cross validation. In view of the fact that the number of acquired images varied across rooms, each room was considered separately during the test experiments. The overall classification rate was then computed as an average, to which the results from each room contributed equally.

Updating Room by Room In the first series of experiments, the system was updated incrementally in a room by room (i.e. class by class) scenario. The system was trained incrementally on one sequence; the corresponding sequence, acquired under roughly similar conditions, was used for testing. The prior-knowledge model was built from one image sequence, acquired under the same illumination conditions and at close time as the training one, but using a different platform. As there are five classes in total, training was performed in 5 steps. In the no-transfer case, the system needed to build the model from scratch, and thus needed to acquire data from at least two classes. In this case, training on each sequence took 4 steps.

Building on top of knowledge acquired from another platform implies a growth in the memory requirements. To evaluate this behavior in relationship to its effects on performance and compare fairly to the system trained without a prior model, we incrementally updated the model without transferred knowledge on another sequence acquired under conditions similar to that of the first training sequence. This experiment makes it possible to evaluate performance and memory growth when both systems are trained on two sequences. The main difference is that in one case both sequences were acquired and processed by the same platform; in the other case, one sequence was acquired and processed by a different platform. We considered different permutations in the rooms order for the updating; for each permutation, we considered 6 different orderings of the sequences used as training, testing, and prior-knowledge sets. Due to space reasons, we report only average results for one permutation, together with standard deviations. Fig. 4a&b report the experimental results obtained at each step, for both systems, including the further steps of the no-transfer system. Fig. 4c&d, give a detailed analysis of the classification rate and the number of stored support vectors obtained at each step of the incremental procedure.

We can see that, for both approaches, the system gradually adapts to its own perception of the environment. It is clear that the knowledge-transfer system has a great advantage in terms of performance over the no-transfer system at the first steps. However, it is interesting to note, that even when both systems have been updated on a full sequence (CR1, Fig 4a), the knowledge-transfer system still maintains an advantage in performance. Considering the differences between the two platforms, and that the transferred knowledge model was built on a single sequence, this is a remarkable result. It can also be observed from Fig. 4d that the SV reduction algorithm facilitated the decay of knowledge from the other platform (in the first incremental step, we did not perform the reduction), while the knowledge acquired by its own sensor gradually becomes the main



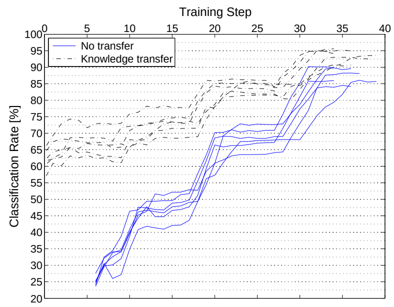
(a) Classification rates at each training step. (b) Number of support vectors at each step.

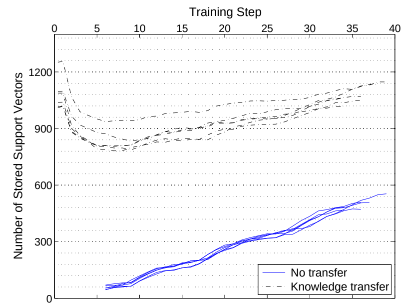
(c) Detailed comparison of the performance of the system at each training step. (d) Detailed comparison of the number of stored support vectors at each step.

Fig. 4. Average results obtained for the system incrementally trained with and without transfer of knowledge in the room by room fashion. Fig. 4a&b compare the final recognition rates and the total number of support vectors for both cases. Fig. 4c&d present a detailed analysis: classification rates obtained for each of the rooms and the amount of support vectors in the final model that originate from the transferred knowledge. In all the plots, the first step "KN." corresponds to the results obtained for the transferred knowledge before any update was performed.

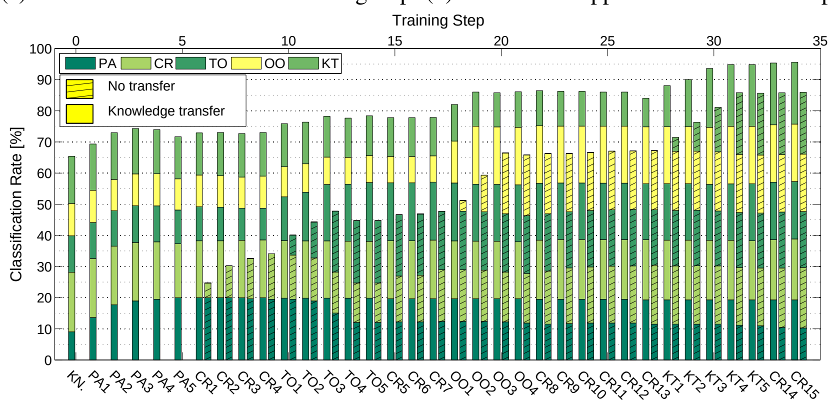
source for the model. As the no-transfer system continued to learn one additional sequence incrementally, its memory growth eventually exceeded the knowledge-transfer case (see Fig 4b). Although the model was built on two sequences acquired by the same platform, the knowledge-transfer system still obtains a comparable performance. We conclude that the transfer of knowledge, in a room by room updating scenario, acts as an effective boosting of performance, without any long-term growth of the memory requirements.

Updating Frames by Frames The second series of experiments explored the behavior of the system in a frames by frames updating scenario, which can be seen as an online incremental learning scenario. Here, for each incremental update, we used a certain number of consecutive frames taken from the training image sequence. Again, the system was trained incrementally on one sequence, and a corresponding sequence was

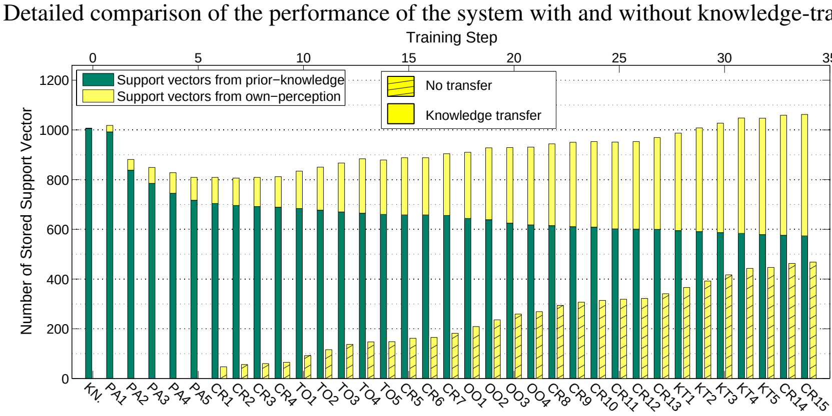




(a) Classification rates at each training step. (b) Number of support vectors at each step.



(c) Detailed comparison of the performance of the system with and without knowledge-transfer.



(d) Number of stored support vectors of incremental experiment with and without knowledge-transfer at each step.

Fig. 5. Average results obtained for the system incrementally trained with and without transfer of knowledge in the frames by frames fashion. Fig. 5a&b compare the final recognition rates and the total number of support vectors for all the experiments. Fig. 4c&d present detailed results for one representative experiment: classification rates obtained for each of the rooms as well as the amount of support vectors in the final model that originate from the transferred knowledge. The labels below each bar indicate the batch of data used for the incremental update. Again, the first step labeled as "KN." corresponds to the results obtained for the transferred knowledge before any update was performed.

used as a test set. We examined the performance of the system for a case when updating was performed using 30 frames per step[4]. Thus, for each experiment, it took more than 30 incremental steps in total to complete a sequence. The prior-knowledge model was built using two complete sequences acquired by the other platform, under the same illumination conditions and very close in time. Again, we benchmarked against the system not using any prior knowledge. In this case, in order to fulfill the requirement of training using at least 2 classes, the first training set consisted of all the images captured in the first room plus the first 30 frames captures in the second room. In consequence, the full training process required five to six less steps than in case of equivalent experiments using the knowledge-transfer scheme. The experiment was repeated 6 times for different orderings of training sequences. Since the number of training steps varied (due to a different number of images in each sequence), we report all the results separately. Fig. 5a&b report the amounts of stored support vectors and classification rates at each step, for all the experiments. This shows the general behavior for both approaches. Fig. 5c&d present results for one of the 6 experiments, so to allow a detailed analysis.

By observing the classification rates obtained at each step in both cases, we see that the advantage of the knowledge-transfer scheme is even more visible here than for the room by room updating scenario. This might be due to the fact that some of the training sets used for the no-transfer case are highly unbalanced. We can observe from Fig. 5c that the performance of the system for previously learned rooms can drop considerably when a new batch of frames is loaded; this is not the case for the knowledge-transfer system. The twelfth step, when the system was updated with frames from the two-persons office (TO3, Fig. 5c), is a typical example. Note that this is a general phenomenon present, although less pronounced, also in the room by room updating scenario. Our interpretation is that the model of the prior-knowledge contains information about the overall distribution of the data. This helps to find a balanced solution when dealing with non-separable instances using soft-margin SVM [11]. As a last remark, we can note that due to use of the SV reduction algorithm the knowledge from the transferred model is gradually removed over time (see Fig. 5d).

Discussion The presented results provide a clear evidence of the advantage of using transferred knowledge across platforms for visual place recognition. The system using the transferred model is able to perform recognition, with a performance well above chance, after having acquired just a small amount of information through its own sensor. By starting to adapt the transferred model to its own perception, performance keeps growing steadily and reaches very high values (above 90%) much earlier than a similar system operating without transferred knowledge. Moreover, adapting a transferred model has shown to pay off considerably in the case of a fast update of the internal representation, where a building from scratch strategy is subject to fluctuations in performance, due to temporary unbalancing of the data. Last but not least, our experiments showed that the SV reduction algorithm, applied before every incremental step, provides an effective way to facilitate the decay of knowledge from the previous platform, as the system learns more about the environment through its own acquisition device.

[4] Experiments conducted for 10 and 50 frames per training step gave analogous results, and for space reasons are not reported here.

6 Conclusion

This paper addressed the problem of visual knowledge transfer across robot platforms. As a starting point on discussing this issue, we considered the case of robots having different characteristics, but engaged in solving the same task. We proposed an framework, based on the SVM classifier, able to adapt the transferred model to the new information in an incremental fashion. Extensive experiments show clearly the effectiveness and promise of our approach.

Future work will focus on adaptation using unlabeled data (semi-supervised scenario), on integrating and updating multiple visual cues and multiple sensors information, and on on-line kernel and feature selection.

Acknowledgments

This work was supported by the the Swedish Research Council contract 2005-3600 - Complex, and by the European IST Project DIRAC (FP6-0027787). This paper only reflects the authors' views and funding agencies are not liable for any use that may be made of the information contained herein.

References

1. Thrun, S., Mitchell, T.: Lifelong robot learning. Robotics and Autonomous Systems 15 (1995)
2. Mitchell, T.: The discipline of machine learning. Technical Report CMU-ML-06-108, CMU (2006)
3. Malak, R.J., J., Khosla, P.K.: A framework for the adaptive transfer of robot skill knowledge using reinforcement learning agents. In: Proc. of ICRA'01
4. Konidaris, G., Barto, A.G.: Autonomous shaping: knowledge transfer in reinforcement learning. In: Proc. of ICML'06
5. Luo, J., Pronobis, A., Caputo, B., Jensfelt, P.: Incremental learning for place recognition in dynamic environments. Technical Report IDIAP-RR-06-52, IDIAP (2006)
6. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of AVS'88
7. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of ICCV'99
8. Wallraven, C., Caputo, B., Graf., A.: Recognition with local features: the kernel recipe. In: Proc. of ICCV'03
9. Linde, O., Lindeberg, T.: Object recognition using composed receptive field histograms of higher dimensionality. In: Proc. of ICPR'04
10. Pronobis, A., Caputo, B., Jensfelt, P., Christensen, H.I.: A discriminative approach to robust visual place recognition. In: Proc. of IROS'06
11. Cristianini, N., Taylor, J.S.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press (2000)
12. Syed, N.A., Liu, H., Sung, K.K.: Incremental learning with support vector machines. In: Proc. of IJCAI'99
13. Downs, T., Gates, K.E., Masters, A.: Exact simplification of support vector solutions. Journal of Machine Learning Research **2** (2002)
14. Luo, J., Pronobis, A., Caputo, B., Jensfelt, P.: The KTH-IDOL2 database. Technical Report 304, KTH, CAS/CVAP (2006) Available at `http://cogvis.nada.kth.se/IDOL2/`.
15. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines. (2001) Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.